Tree Congestion for Complete n-Partite Graphs

Allen Cox *†

August 24, 2007

Abstract

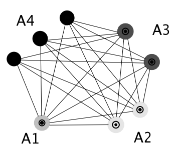
The tree congestion and number of edge disjoint paths existing in n-partite graphs is explored, along with the spanning tree congestion for such graphs.

1 Introduction

A graph G consists of a set of vertices, V_G, and a set of edges, E_G. Each edge represents a connection between two vertices. We will only be using simple graphs, or graphs where multiple edges do not connect the same two vertices, and no edges exist such that the endpoints of the edge are both a single vertex. A vertex v is *adjacent* to a vertex u if there exists an edge between them. An edge g that has an endpoint of v is *incident* to v.

A complete graph is a graph K_n with n vertices such that each vertex in K_n is connected to every other vertex. A complete bipartite graph, $K_{m,n}$, is a graph with two sets of vertices M and N such that $|M| = m$ and $|N| = n$, with every vertex in M adjacent to each vertex in N, and with no edges of the same set being adjacent. To take this further, a complete n-partite graph, $K_{a_1,a_2,\ldots,a_n}$, $a_1 \leq a_2 \leq \ldots \leq a_n$, is a graph that contains n sets of vertices, $A_1, A_2, \ldots, A_n$, with $|A_i| = a_i$ and each vertex in A_i is adjacent to every vertex v such that $v \notin A_i$ for $1 \leq i \leq n$.

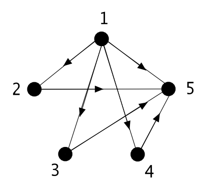
Figure 1: A Complete 4-Partite Graph



*Kent State University
†This work was jointly supported by California State University, San Bernardino, and NSF-REU Grant DMS-0453605.

Tucker [7] defines that for a graph G, a *path* is a sequence of distinct vertices $(x_1, x_2, \dots, x_n)$ such that consecutive vertices are adjacent. A graph is a *tree* if there exists only one path between a pair of vertices for all such distinct pairs in the graph, and that the number of edges in a graph is equal to the number of vertices minus one, or $|E_G| = |V_G| - 1$. The *diameter* of a tree is the length of the longest path in the tree, where the length of a path $(x_1, x_2, \dots, x_n)$ would be $n-1$, for the number of edges contained in the path. Also, a graph is called a *connected* graph if there exist paths between all pairs of vertices in a graph. The maximal number of edge disjoint paths for a graph G, denoted m_G, is the maximal number of paths between two vertices $u, v \in V_G$ that share no common edge, among all possible u, v. The concept of edge disjoint paths is illustrated in Figure 2.

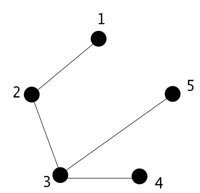
Figure 2: Edge Disjoint Paths Between Vertices 1 and 5 for K_5



Note that m_G would correspondingly be 4. It is not possible for $m_G \geq deg(G)$ in any case, because as the paths are edge disjoint, we cannot make more edge disjoint paths from a vertex than we have edges connected to a vertex.

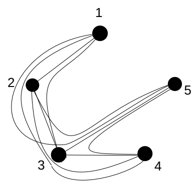
For G, there exist a number of trees, denoted $T_1, T_2, \dots, T_n$ which may be embedded onto G, with the only conditions being that all vertices in G are used in any such T_i, and that T_i meets the conditions of a tree. For a tree of a graph G to be called a spanning tree, the additional condition must hold that for any edge in the spanning tree, the same edge exists in the original graph G.

Figure 3: A Spanning Tree for K_5



Ostrovskii defines what is called an $H - Layout$ L of G, but for now, we only need to use a more specfic form of the following definition. For an embedded tree on G, denoted T, the $T - Layout$ L of G is a set $\{P_g : g \in E_G\}$ of paths P_g in T that join the endpoints of edge g. For example, a $T - Layout\ L$ of K_5, with T equal to our spanning tree in Figure 3 would equal the set $L = \{(1,2),(1,2,3),(1,2,3,4),(1,2,3,5),(2,3),(2,3,4),(2,3,5),(3,5),(3,4),(4,3,5)\}$. The paths each represent a connection that occurs in the original graph, such as the path $(2,3,5)$ representing the edge between vertices 2 and 5 in the original graph. The T-Layout is shown visually in Figure 4 for our spanning tree in Figure 3.

Figure 4: T-Layout of K_5



Next, the following definitions come directly from Ostrovskii [6], but have been modified specifically for tree embeddings:

Definition 1. *For an edge t in T the congestion of t is the number of times t appears in a $T - Layout\ L$ of G, or simply $cs(t, L) = |\{P_g \in L : t \in P_g\}|$.*

Definition 2. *For a tree T with edge set E_T embedded onto a graph G, the congestion of T is*

$$c(T) = \max_{t \in E_T} cs(t, L)$$

Definition 3. *The tree congestion of a graph G denoted $t(G)$ is the minimum congestion of all possible tree embeddings.*

Definition 4. *The spanning tree congestion of a graph G denoted $s(G)$ is the minimum congestion of all possible spanning tree embeddings.*

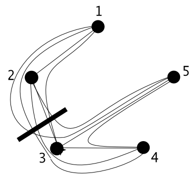
Now we have several definitions from Ostrovskii that will provide the main ideas for work to follow. Congestion can also be thought of as *tree cutwidth*, which can be interperted visually on a tree by contructing paths from our $T - Layout\ L$ of G set on the tree, and then making a cut, or intersection, between two vertices and counting the number of paths intersected. This cut gives the congestion of that edge, and the maximum of all possible cuts would give the congestion of the tree.

In Figure 5 is a representation of spanning tree congestion, from our spanning tree in Figure 3. The thick black line represents the cut.

The cut between vertices 2 and 3 intersects six edges, which is the maximum for the graph, hence by definition, the congestion of the spanning tree is six.

A theorem due to Ostrovskii [6] is already established that relates m_g, $t(G)$, and $s(G)$. It will be used frequently for later results.

Figure 5: T-Layout of K_5, with an example of a cut

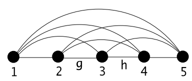


Theorem (Ostrovskii [6]). *For a connected graph, G, $m_G = t(G) \leq s(G) \leq |E_G| - |V_G| + 2$.*

2 Background

The idea of congestion or cutwidth has frequently been researched for embeddings of a graph G onto a graph other than a tree. The cutwidths of G have also been explored for linear embeddings and cyclic embeddings. To define these terms, we will use a slightly different form of the definitions previously used to define tree congestion. An $H - Layout$ L of a graph G is a set $\{P_g : g \in E_G\}$ of paths P_g in H that join the endpoints of edge g, with the cutwidth of an edge g in defined to be the number of times that g appears in this set. H can be referred to as our *host graph*. A linear embedding of G can be constructed by taking the vertices in G, and arranging them along a line, with edges made between vertices connected in G. For a cyclic embedding of G, a similar approach is taken, except with an arrangement of the vertices in G in a cycle. The cutwidth of the embedding is given by the maximum cutwidth of the edges, and the cutwidth of G is given by the minimum cutwidth of all possible embeddings.

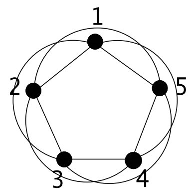
Figure 6: A Linear Embedding of K_5



Denote the linear embedding above to be L_1 and the cyclic embedding to be C_1. The linear cutwidth of L_1, or $lcw(L_1)$ is equal to the maximum edge cutwidth, which is six for edges g and h. The cyclic cutwidth of C_1, or $ccw(C_1)$, is three, as each edge has the same cutwidth.

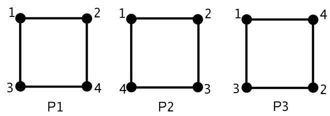
Note that there is more than one possible embedding for a given graph G. The linear and cyclic embedding of a graph will be different for each possible labeling of vertices, as our embeddings are made based on the consecutive numbering of the vertices. For K_5, we have a complete graph, and our labeling is inconsequential as every vertex is adjacent to every other vertex. For a

Figure 7: A Cyclic Embedding of K_5



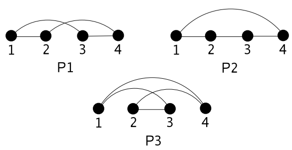
connected square, or 2-cube, where every vertex is not adjacent to every other vertex, there exist several different labelings up to isomorphism. We will consider a labeling to be non-isomorphic if the labeling cannot be obtained from a rotation or flip of the graph. More formally, non-isomorphic labelings are labelings which have different vertex adjancencies.

Figure 8: Graph Labelings of a 2-Cube



Each one of these labelings will produce different linear and cyclic embeddings, and therefore also have different cutwidths. Recall that the linear cutwidth of G is found by taking the minimum of the cutwidths for all possible embeddings. The cyclic cutwidth of G is found in a similar manner.

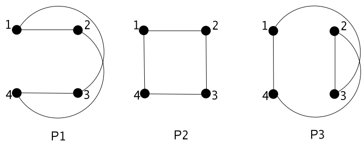
Figure 9: Linear Embeddings of the 2-Cube



Note that $lcw(G) = 2$, as we have $lcw(P1) = 2$, $lcw(P2) = 2$, and $lcw(P3) = 4$. The linear cutwidth of G is the minimum of these values. Correspondingly, $ccw(G) = 1$.

In a paper by Chavez and Trapp [2], it was established that if G is a tree, then $lcw(G) = ccw(G)$. Also, it was proved by Johnson [5] that for a complete

Figure 10: Cyclic Embeddings of the 2-Cube



bipartite graph $K_{m,n}$,

$$lcw(K_{m,n}) = \begin{cases} \frac{mn}{2} & \text{if } mn \text{ is even;} \\ \frac{mn+1}{2} & \text{if } mn \text{ is odd.} \end{cases}$$

Also proven were partial results for cyclic cutwidth. For $K_{m,n}$, it was shown that

$$ccw(K_{m,n}) = \begin{cases} \frac{mn}{4} & \text{if } m, n \text{ both even;} \\ \frac{mn+3}{4} & \text{if } m, n \text{ both odd and } m = n. \end{cases}$$

Note that the congestion of a tree can equivalently be described as the cutwidth of a tree based on the definition of cutwidth. For bipartite graphs and tripartite graphs, the maximal number of edge disjoint paths and tree congestion were explored in research projects by Stephen Hruska [4] and Diana Carr [1], respectively. Recall that m_G is the maximal number of edge disjoint paths between two vertices in a graph G over all such distinct pairs. Also, $t(G)$ is the minimum congestion of all possible tree embeddings of a graph G, and $s(G)$ is the minimum congestion of all possible spanning tree embeddings. Hruska investigated the tree congestion of bipartite graphs. Carr took his results one step further, and investigated the tree congestion for tripartite graphs. The results of each are displayed below, with both using the previous theorem from Ostrovskii [6] for $m_G = t(G) \leq s(G)$.

Theorem (Hruska [4]). *For $G = K_{m,n}$, $m \leq n$, $m_G = t(G) = s(G) = 1$ if $m = 1$, and $m_G = t(G) = n$, $s(G) = m + n - 2$ if $m \geq 2$.*

Theorem (Carr[1]). *For $G = K_{m,n,l}$, $m \leq n \leq l$,*

$$m_G = t(G) = \begin{cases} l + 1 & \textit{if } m = 1; \\ n + 1 & \textit{if } m \geq 2. \end{cases}$$

$$s(G) = \begin{cases} l + 1 & \textit{if } m = 1; \\ (2m + n + l) - 2 & \textit{if } m \geq 2. \end{cases}$$

These results will be expanded to n-partite graphs.

3 Main Results

3.1 $t(G)$ for Complete n-Partite Graphs

A complete n-partite graph, $K_{a_1,a_2,\ldots,a_n}$, $a_1 \leq a_2 \leq \ldots \leq a_n$, is a graph that contains n sets of vertices, $A_1, A_2, \ldots, A_n$, with $|A_i| = a_i$ and each vertex in A_i is connected to every vertex v such that $v \notin A_i$ for $1 \leq i \leq n$. Also, we have by Ostrovskii [6] that $m_G = t(G)$, so by establishing m_G we are given $t(G)$. For $n = 2$, refer to Hruska [4], and for $n = 3$, refer to Carr [1]. Their previous results will be assumed.

Theorem 1. *For $G = K_{a_1,a_2,\ldots,a_n}$, with $a_1 \leq a_2 \leq \ldots \leq a_n$, $|A_1| = a_1, |A_2| = a_2, \ldots, |A_n| = a_n$, and $n \geq 4$,*

$$m_G = t(G) = \begin{cases} 1 + a_3 + a_4 + \cdots + a_n & \textit{if } a_1 = 1; \\ a_2 + a_3 + \cdots + a_n & \textit{if } a_1 \geq 2. \end{cases}$$

3.1.1 Case 1: $a_1 = 1$

Recall that m_G is the the maximal number of edge disjoint paths between all distinct pairs of vertices u, v in a graph G. To find our maximal edge disjoint paths, we must choose the u and v with the highest possible degrees, as we will be bounded by the lesser of these two degrees. Since G is a complete n-partite graph, each vertex will have degree equal to the sum of the cardinalities of the sets it is not contained in. Knowing that $a_1 \leq a_2 \leq \ldots \leq a_n$, a vertex in A_1 will have the highest degree of $a_2 + a_3 + \cdots + a_n$. Since u and v cannot both be chosen in A_1 since $|A_1| = 1$, at least one must be chosen to be in some set not equal to A_1. The next highest degree belongs to any vertex of A_2, with degree equal to $a_1 + a_3 + \cdots + a_n$, thus giving an upper bound for m_G. Hence u and v chosen in A_1 and A_2 provides the highest possible minimum degree for all such pairs u, v. With $u \in A_1$, $v \in A_2$, we have a_3 paths of length two of the form uy_3v, for all $y_3 \in A_3$, a_4 paths of length two of the form uy_4v,for all $y_4 \in A_4$, and continued up to a_n paths of length two of the form uy_nv, for all $y_n \in A_n$. With our direct path uv, this gives us a total of $1+a_3+a_4+\cdots+a_n$ edge disjoint paths, which is our upper bound. Thus $m_G = t(G) = 1 + a_3 + a_4 + \cdots + a_n$.

3.1.2 Case 2: $a_1 \geq 2$

In this case, note that u and v can be chosen to both have degree $a_2+a_3+\cdots+a_n$, hence $m_G \leq a_2 + a_3 + \cdots + a_n$. Choose $u, v \in A_1$, which gives us a_2 paths of length two of the form uj_2v, for $j_2 \in A_2$, a_3 paths of length two of the form uj_3v, for $j_3 \in A_3$, and continued up to a_n paths of length two of the form uj_nv for $j_n \in A_n$, for a total of $a_2 + a_3 + \cdots + a_n$ paths. This gives us our upper bound of edge disjoint paths, hence $m_G = t(G) = a_2 + a_3 + \cdots + a_n$.

3.2 $s(G)$ for Complete n-Partite Graphs

For our complete n-partite graph G, the spanning tree congestion, denoted $s(G)$, is the minimum congestion of all possible spanning tree embeddings. By Ostrovskii [6], $s(G)$ has a lower bound of $t(G) = m_G$. Again, for $n = 2$, refer to Hruska [4], and for $n = 3$, refer to Carr [1]. Their previous results will be assumed.

Theorem 2. *For $G = K_{a_1,a_2,\ldots,a_n}$, with $a_1 \leq a_2 \leq \ldots \leq a_n$, $|A_1| = a_1, |A_2| = a_2, \ldots, |A_n| = a_n$, and $n \geq 4$.*

$$s(G) = \begin{cases} 1 + a_3 + a_4 + \cdots + a_n & \text{if } a_1 = 1; \\ 2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2 & \text{if } a_1 \geq 2. \end{cases}$$

3.2.1 Case 1: $a_1 = 1$

With $|A_1| = 1$, construct a star with the only vertex $v \in A_1$ as the center with all other vertices connected to v. Since G is an n-partite graph, v is connected to all vertices not in A_1, which is in this case all vertices but v. These connections in the star already exist in G, and since we are using all the vertices in G, we have a spanning tree. The only way to traverse to all other vertices from some vertex $a_i \neq v$ is to travel from a_i through v to each vertex that a_i connects to. This will result in a number of paths traversing $a_i v$ equal to the sum of the cardinalities of each set A_j such that $i \neq j$, which includes the direct path $a_i v$. This congestion for an edge $a_i v$ can be represented more succinctly by

$$cs(a_i v) = \sum_{i \neq j} |A_j|.$$

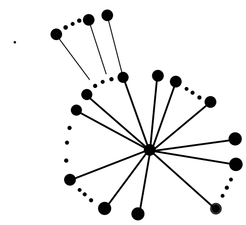
Since we are looking for the maximal edge congestion, this can be found be picking any edge connecting v to a vertex in a_2, which gives our least cardinality a coefficient of zero in our summation. Thus our maximal edge congestion is $1 + a_3 + a_4 + \cdots + a_n$, which is the congestion of the tree. Since this is equal to $t(G)$, our lower bound for $s(G)$, we have found the minimal spanning tree congestion for this case.

3.2.2 Case 2: $a_1 \geq 2$

First note that a tree of diameter two will not be applicable in this situation. Say $v_1 \in A_k$ is picked as the center of the star as in the tree for case one. Since $a_k \geq a_1 \geq 2$, there will exist some $v_2 \in A_k$ such that $v_2 \neq v_1$. We are constructing a spanning tree, so it is not valid to connect v_2 to v_1. Hence there must exist some vertex w between v_1 and v_2, which invalidates using our star diagram. Thus we must extend our tree to diameter three or more. There are two possible subgraphs of diameter three that our tree must either be equal to or contain. The first case is for graphs that are considered *extended stars*.

An *extended star* E_S is a spanning tree for a complete n-partite graph $G = K_{a_1,a_2,\ldots,a_n}$ such that there exists a central vertex $\alpha \in A_i$ connected to all vertices $v \notin A_i$, and each edge constructed so far is of the form αv. The remaining $\alpha' \in A_i$ such that $\alpha \neq \alpha'$ are then connected to vertices $v \notin A_i$, with each v and α' pairing distinct.

Figure 11: An Extended Star



Now we look to minimize our maximal edge congestion for this specific construction towards the goal of finding a value for $s(G)$. It is clear that the edges with the most congestion will be edges of the form $v\alpha$ such that v also connects to a vertex α' as defined above. These edges will not only be traversed for paths starting from vertex v, but also paths starting from vertex α'. Thus an edge $v\alpha$ for $v \in A_b$, $\alpha \in A_i$, $b \neq i$ will have $\left(\sum_{j=1}^{n} a_j\right) - 1 - a_b$ paths originating from v that traverse through $v\alpha$. The subtraction of at the end is to account for the one α' that is connected to v, thus not using edge $v\alpha$ for this path, and also to account for the members of A_b that $v \in A_b$ will not make paths to. Then, $v\alpha$ will also have the congestion from paths originating from vertex α' for $\alpha' \in A_i$ which similarly will be $(\sum_{k=1}^{n} a_k) - 1 - a_i$. More formally, this gives us

$$cs(v\alpha) = \left(\sum_{\substack{1 \leq j \leq n \\ b \neq j}} a_j + \sum_{\substack{1 \leq k \leq n \\ i \neq k}} a_k \right) - 2.$$

Note that when our two summations are added, all of our terms will have a coefficient of two except for a_b and a_i. This is because a_b is not added in the first summation, and a_i is not added in the second summation, but all other cardinalities appear once in each sum, thus giving us coefficients of one for a_b and a_i, and coefficients of two for all other terms.

The congestion of an edge of the form $d\alpha$ such that $d \in A_j$ is not also connected to an α' vertex is

$$\sum_{\substack{1 \leq i \leq n \\ j \neq i}} a_i.$$

This is to account for the paths that will be made to all vertices except those vertices in A_j. Therefore, we must show that this is less than our congestion for $v\alpha$, which can be done by showing $2\sum_{x=1}^{n} a_x - (2 + a_b + a_i) \geq \sum_{x=1}^{n} a_x - a_j$.

This is equivalent to showing

$$\begin{array}{ll} \sum_{x=1}^{n} a_x - a_b \geq 2 & \text{if } A_i = A_j, \\ \sum_{x=1}^{n} a_x - a_i \geq 2 & \text{if } A_b = A_j, \\ \sum_{x=1}^{n} a_x - a_b - a_i \geq 2 & \text{if } A_i \neq A_j \text{ and } A_b \neq A_j. \end{array}$$

Since $n \geq 4$, and $a_i \geq a_1 \geq 2$ for $1 \leq i \leq n$, then our above three summations will each have at least four terms, and each one of those terms will be greater than or equal to two. Even if one or two terms are subtracted as above, our sum is still greater than two as there will exist at least two terms on the left, each at least equal to two. So the inequality is established, and we may now focus on the edges of larger congestion of the form $v\alpha$.

We have that edges of the form $v\alpha$ such that v connects to some α' vertex have maximal congestion for an extended star graph, and now we need to minimize this maximal value to get the congestion for our original graph G.

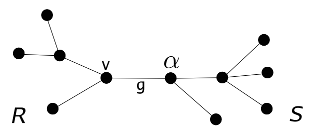
In our summation, we have two coefficients that are equal to one, with all other coefficients equal to two. The coefficients of one correspond to the sets that v and α are in. Therefore, we need to make sure that the sets A_b and A_i are chosen to be as large as possible, to avoid a large cardinality set having a coefficient of two when it is not neccessary. We know that $a_1 \leq a_2 \leq \ldots \leq a_n$. Let $b = n$ and $i = n - 1$. Hence by our contruction all remaining $\alpha' \in A_i = A_{n-1}$ not equal to α will be connected to distinct vertices in $A_n = A_b$, as $a_n \geq a_{n-1}$. Thus we have a maximal edge congestion for our extended star as $2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2$. We can only modify which two cardinalities have a coefficient of one instead of two. These cardinalities have been chosen as the largest, thus minimizing our maximum edge congestion, and hence gives us the congestion of our extended star spanning tree as $2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2$. We will denote *optimal* extended stars as an extended star E'_S having this minimal congestion. Since our construction was generalized, an optimal extended star can be embedded on any complete n-partite graph G.

For our extended star, we had vertices v that were connected to a central vertex α with some v each connecting to one additional $\alpha' \neq \alpha$. We have not explored what occurs in terms of congestion when a vertex v is connected to multiple vertices besides α and α', if our vertex α' is connected to additional vertices, or when both occur together. Denote edge g to have endpoints of v and α such that (i), v is connected either to at least two additional vertices, or (ii), v has a path of length two or more adjoined to it that does not include the vertex α.

In either case, let edge g with endpoints v and α partition the graph into two sets of vertices, R and S. R will consist of the set of vertices r such that each r is contained in a path that traverses to v without traversing through vertex α, and S will consist of the vertices s such that each s is contained in a path that traverses to α without traversing vertex v. Also, $v \in R$, and $\alpha \in S$. This can be also thought of as adjoining two trees of three or more vertices by a single edge g. If $|R|$ or $|S|$ is less than three, then the set consists only one vertex connected to v or α, which gives us an extended star. Thus assume $|R|$

and $|S| \geq 3$, and furthermore that R and S are disjoint subsets of V_{T_G} such that $R \cup S = a_1 + a_2 + \cdots + a_n$ and our construction yields a spanning tree for our complete n-partite graph G. Trees that follow the above critiria will be denoted as *divided* trees.

Figure 12: A Divided Tree



For the edge g, let β_i represent the number of vertices from A_i that are in set R, with $\beta_i \geq 0$, $1 \leq i \leq n$, and $\beta_1 + \beta_2 + \cdots + \beta_n = |R|$. Therefore, each vertex of A_i that is contained in R will have to connect to $|S| - (a_i - \beta_i)$ vertices that require using edge g in a path. $a_i - \beta_i$ is the number of vertices in A_i not in R, or the number of vertices in A_i in S.

Therefore g will have an edge congestion of

$$\sum_{\substack{1 \leq j \leq n \\ j \neq i}} \beta_j(|S| - a_j + \beta_j)).$$

Next we will show that this edge congestion is greater than that of our extended star, or more formally, for an edge g as described above in our divided tree, and an optimal extended star E_S embedded onto the same complete n-partite graph G, $cs(g) \geq c(E_S)$, or

$$cs(g) = \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \beta_j(|S| - a_j + \beta_j) \geq 2(a_1 + a_2 + \cdots + 2a_{n-2}) + a_{n-1} + a_n - 2.$$

We will induct on the number of vertices in V_G. Note that neither R nor S can have cardinality lower than three as this would result in an extended star or a star, and neither R nor S can consist of only vertices from a distinct A_i for $1 \leq i \leq n$, as this would require edges connecting vertices not connected in our graph G, thus invalidating a spanning tree. Without loss of generality, begin with $|R| = 3$. For $R = \{\alpha, \gamma, \delta\}$, for $\alpha \in A_j, \gamma \in A_k$, and $\delta \in A_\ell$ for $1 \leq j, k, \ell \leq n$. This will result in $3(a_1 + a_2 + \cdots + a_n - 3) - (a_j - 1) - (a_k - 1) - (a_\ell - 1)$ paths through edge g. If two of our vertices in R are both in A_j,A_k, or A_ℓ, then this sets corresponding cardinality will have a coefficient of one in our sum above. If none of our vertices in R are in the same set, then a_j,a_k, and a_ℓ will all have coefficients of two in the sum. Regardless, this will still result in a congestion for g greater than that of our optimal extended star, which was $2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2$.

Assume for $|V_G| = m$, $|R| = k$ and $|S| = m - k$, with $|R|, |S| \geq 3$, that the congestion for edge g is greater than the congestion of our optimal extended star. This gives us

$$\sum_{\substack{1\leq j\leq n\\ j\neq i}} \beta_j(|S|-(a_j-\beta_j)) \geq 2(a_1+a_2+\cdots+2a_{n-2})+a_{n-1}+a_n-2$$

which can be enumerated as

$$\beta_1(|S|-(a_1-\beta_1)+\beta_2(|S|-(a_2-\beta_2)+\cdots+\beta_n(|S|-(a_n-\beta_n)$$
$$\geq 2(a_1+a_2+\cdots+a_{n-2})+a_{n-1}+a_n-2.$$

Now we need to prove for $|R|=k+1$ that the result still holds. Note that a vertex *not* in our original graph G is being adjoined to set R in our divided tree. For $|R|=k+1$, adding a vertex in R will produce two cases. The first case is if a vertex $\omega \in A_w$ is adjoined to R such that no other vertices $\omega' \in A_w$ exist in R, and the second case is if a vertex $\eta \in A_q$ is adjoined to R such that at least one other vertex in R is contained in A_q.

For the first case, another term will be in our summation on the left to account for the $\omega \in A_w$ being adjoined to T. We will have an additional $\beta_w(|S|-(a_w-\beta_w))$ on the left, and on the right we will have an additional constant of at most two, depending on whether $w<n-1$. We already know that

$$\beta_1(|S|-(a_1-\beta_1)+\beta_2(|S|-(a_2-\beta_2)+\cdots+\beta_n(|S|-(a_n-\beta_n)$$
$$\geq 2(a_1+a_2+\cdots+a_{n-2})+a_{n-1}+a_n-2,$$

but we need to establish that $\beta_w(|S|-(a_w-\beta_w))\geq 2$ to verify that the additional terms on the left outnumber the additional terms on the right. Since no other vertices in A_w exist in R, $\beta_w=1$. Hence we have $\beta_w(|S|-(a_w-\beta_w)) = |S|-a_w+1$, and since S must contain A_w except for the vertex adjoined to R, and cannot be equal to the remainder of A_w by construction of our tree, then $|S|>a_w$. Hence $|S|-a_w\geq 1$, and $|S|-a_w+1\geq 2$. So our result still holds.

The second case is if a vertex $\eta\in A_q$ is adjoined to R such that there exists at least one other vertex $\eta'\in A_q\cap R$, with $\eta'\neq q$. Therefore in our summation

$$\sum_{1\leq j\leq n} \beta_j(|S|-(a_i-\beta_j)),$$

our β_q variable will be replaced by β_q+1. Hence in our sum, we will have $(\beta_q+1)(|S|-(a_q-\beta_q+1)) = \beta_q|S|-\beta_q a_q+\beta_q^2+\beta_q+|S|-a_q+\beta_q+1$ instead of $\beta_q|S|-\beta_q a_q+\beta_q^2$. Therefore our left side increases by the additional terms $|S|-a_q+2\beta_q+1$, and our right side increases by two, for our a_q being replaced by a_q+1 in $2(a_1+a_2+\cdots+a_{n-2})+a_{n-1}+a_n-2$. Hence we must verify that $|S|-a_q+2\beta_q+1\geq 2$, or equivalently $|S|-a_q+2\beta_q>0$. We know that $a_q-\beta_q$ is the number of vertices in A_q that are also in S, and hence will not be negative, so $a_q-\beta_q\geq 0$. We know that $|S|>a_q$, as explained in our previous case, hence $|S| - a_q \geq 1$, and with $\beta_q \geq 1$ by our case statement, then $|S| - a_q + 2\beta_q > 0$. Therefore, our result holds for the second case. By proof of mathematical induction,

$$cs(g) = \beta_1(|S| - (a_1 - \beta_1) + \beta_2(|S| - (a_2 - \beta_2) + \cdots + \beta_n(|S| - (a_n - \beta_n)$$
$$\geq 2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2.$$

Recall what we have shown. First, we established the congestion of an optimal extended star E'_S to be $2(a_1+a_2+\cdots+a_{n-2})+a_{n-1}+a_n-2$. Next we established for a spanning tree not in the form of an extended star, denoted as a divided tree, that the congestion of the specific edge g was always more than the congestion of our extended star. Furthermore, by definition of congestion for edges and trees and with any divided tree represented by D, if $cs(g) \geq c(E'_S)$, then $c(D) \geq c(E'_S)$. Since the spanning tree congestion of a graph G is the minimum congestion of all possible spanning trees, then for $G = K_{a_1,a_2,\ldots,a_n}$ with $a_1 \geq 2$ and $n \geq 4$, $s(G) = 2(a_1 + a_2 + \cdots + a_{n-2}) + a_{n-1} + a_n - 2$.

3.3 $|E_G|$ and $|V_G|$ for Complete n-Partite Graphs

Recall that $t(G)$ and $s(G)$ are bounded above by $|E_G| - |V_G| + 2$, by Ostrovskii.

Theorem 3. *For $G = K_{a_1,a_2,\ldots,a_n}$ with $a_1 \leq a_2 \leq \ldots \leq a_n$ and $|A_i| = a_i$ for $1 \leq i \leq n$,*

$$|E_G| - |V_G| + 2 = \sum_{k=1}^{n-1} a_k(a_{k+1} + a_{k+2} + \cdots + a_n) - \sum_{j=1}^{n} a_j + 2.$$

The proof is quite clear. For any vertex in a_1, it must connect to $a_2 + a_3 + \cdots + a_n$ vertices. For a_2, it must connect to all vertices except ones in its own set and the vertices in a_1 it has already been connected to, or $(a_3 + a_4 + \cdots + a_n)$. This continues up to any vertex in a_{n-1} having to connect to a_n vertices, thus giving us$\sum_{k=1}^{n-1} a_k(a_{k+1}+a_{k+2}+\cdots+a_n)$ edges. The number of vertices is found by adding the cardinalities of the vertex sets, or $a_1 + a_2 + \cdots + a_n = \sum_{j=1}^{n} a_j$. This gives our result.

4 Future Research

The work I have done for complete n-partite graphs could be extended to additional areas. Johnson [5] has researched the linear and cyclic cutwidths of complete bipartite graphs, and Hartung [3] has done work on the wirelength of complete bipartite graphs, with wirelength being the sum of the distances between points on a linear embedding of a graph. Just as $t(G)$ and $s(G)$ was expanded from complete bipartite and tripartite graphs to complete n-partite graphs in this paper, finding the linear and cyclic cutwidths and wirelength for complete n-partite graphs is a fitting extension for additional research.

5 Acknowledgments

Since this research was done at the REU program at California State University, San Bernardino, I would like to first thank Dr. Joseph Chavez and Dr. Rolland Trapp for their organization of the program, and for giving me the opportunity to participate. I would also like to thank both of them for their continued input and guidance throughout the program. Thanks is also due to the other REU participants for the memorable experiences we enjoyed during the summer, and to those close to me for their continuing encouragment and support.

References

[1] Carr, Diana, The Tree Congestion of Graphs, REU Project, California State Univ. San Bernardino, 2005.

[2] Chavez, J.D. and Trapp, R., The Cyclic Cutwidth of Trees, *Discrete Applied Mathematics*, 87 (1998) 25-32.

[3] Hartung, Elizabeth J., The Linear and Cyclic Wirelength of Complete Bipartite Graphs, REU project, California State Univ. San Bernardino, 2004.

[4] Hruska, S., On Tree Congestion, REU Project, California State Univ. San Bernardino, 2004.

[5] Johnson, Matt, The Linear and Cyclic Cutwidth of the Complete Bipartite Graph, REU Project, California State Univ. San Bernardino, 2003.

[6] Ostrovskii, M.I., Minimal congestion trees, *Discrete Mathematics*, 285 (2004) 219-226.

[7] Tucker, Alan, "Applied Combinatorics", John Wiley and Sons, Inc., 1995.