The Tree Congestion of Graphs

Diana Carr

August 24, 2005

Abstract

Edge congestion can be thought of as the cutwidth of a graph. In this paper we embed complete tripartite graphs into trees and spanning trees and determine the tree congestion and the spanning tree congestion. Considering a known theorem relating detours, tree congestion, and spanning tree congestion, we summarize results calculated for trees, complete bipartite graphs, and grids. In addition, we investigate the congestion for other families of graphs.

1 Introduction

A graph, G, consists of a set, V, of vertices and a set, E, of edges that join pairs of distinct vertices together. A graph in which every vertex is connected to every other vertex is called complete. A complete bipartite graph, $K_{m,n}$, consists of two disjoint sets of vertices, M and N, such that every vertex in M is joined by an edge to every vertex in N, where $|M| = m$ and $|N| = n$. A complete tripartite graph $K_{m,n,l}$ contains three disjoint sets of vertices, M,N, and L, with $|M| = m, |N| = n,$ and $|L| = l$, such that every vertex in M is joined by an edge to every vertex in N, every vertex in M is joined by an edge to every vertex in L, and every vertex in N is joined by an edge to every vertex in L. Figure 1 shows an example of $K_{3,4}$ and Figure 2 shows an example of $K_{1,2,2}$.

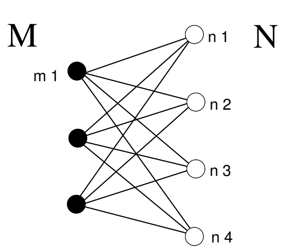


Figure 1: $K_{3,4}$

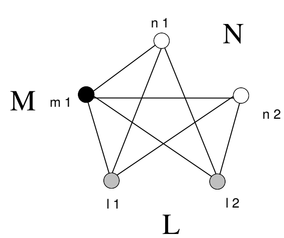


Figure 2: $K_{1,2,2}$

A *linear embedding* of a graph, G, is a representation of G with all of the vertices and edges of G embedded onto a line. All edges that connect vertices in the non-linear embedding of G also connect vertices in the linear embedding. Figure 3 shows an example of a linear embedding of a $K_{3,4}$.

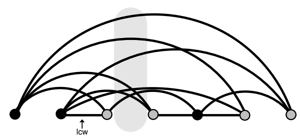


Figure 3: Linear Embedding of $K_{3,4}$

A *cyclic embedding* of a graph, G, is a representation of G in which all of the vertices of G are embedded onto a cycle. All edges that connect vertices in the original representation of G also connect the same vertices in the cyclic embedding of G. The cyclic embedding of $K_{3,4}$ is shown in Figure 4.

The cut of a region is the number of edges that cross the region between two adjacent verticies in the graph. The cut of the shaded region in Figure 3 is seven whereas the cut of the shaded region in Figure 4 is three. The maximum cut of an embedding of a graph is the largest cut that occurs on the graph. Notice that the linear cutwidth, denoted *lcw*, in Figure 3 is eight and the cyclic cutwidth, denoted *ccw*, in Figure 4 is five. The minimum of all of the possible maximum cuts over all of the possible embeddings is called the *cutwidth of G*.

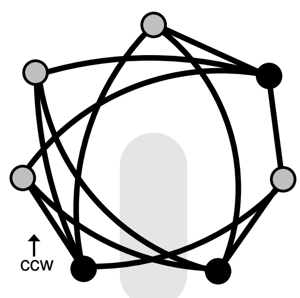


Figure 4: Cyclic Embedding of $K_{3,4}$

We shall now define the important terms and ideas in this paper. Let G denote a connected graph in which E_G represents set the of edges and V_G represents the set of vertices. Let T denote a tree such that $V_T = V_G$. A *spanning tree* of a connected graph, G, is a tree, T, where $V_T = V_G$ and $E_T \subseteq E_G$. A *path* in G is a sequence of distinct vertices $(x_i, x_{i+1}, ..., x_{i+j})$ in which consecutive vertices are connected in G.

Let u and v be vertices in G. Two paths connecting u and v are *edge-disjoint* if they share no common edges. The *maximal number of edge-disjoint paths* connecting u and v in G is denoted as $m(u, v)$. Considering all possible pairs of vertices, we determine the maximum over every $m(u, v)$ and define it as m_G:

$$m_G = \max\{m(u, v)|u, v \in V_G\}.$$

For example, consider K_6. Figure 5 shows a maximum of five edge-disjoint paths in K_6 between vertices 1 and 4, so $m(1, 4) = 5$. There is no vertex with degree higher than five, therefore we would never be able to achieve a maximal edge-disjoint path higher than this. Thus, $m_G = 5$.

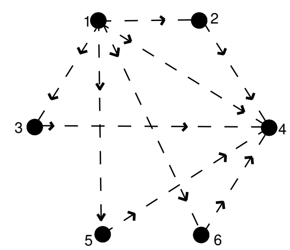


Figure 5: K_6

Given $G = (V_G, E_G)$, let T be a tree such that $V_T = V_G$. The set of all trees such that $V_T = V_G$ is denoted T_G. A tree in T_G is called a *spanning tree* if it is a subgraph of G and connects all the vertices together with $E_T \subseteq E_G$. Let S_G denote the set of all spanning trees of G.

Each edge $e = (a, b)$ in G corresponds to a path P_e in T connecting the same pair of vertices, a and b. We call these paths *detours*, and denote the set of detours as L. For a particular edge g in T, we obtain the *congestion* of g,

$$c(g, T) = |\{P_g \in L : g \in P_g\}|,$$

by counting the number of detours that g appears in. The *congestion of G embedded into T* is defined as the maximum $c(g, T)$ over every edge g in T. More formally,

$$c(G : T) = \max\{c(g, T)|g \in E_T\}.$$

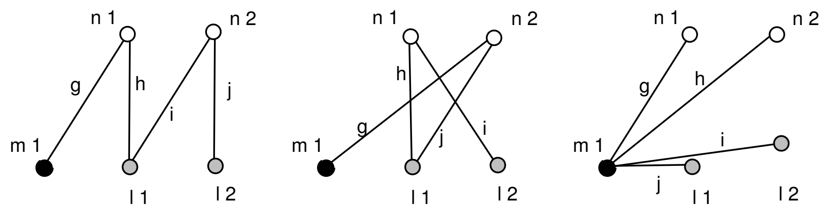
This paper examines the *tree congestion of G* and the *spanning tree congestion of G*. Tree congestion can basically be thought of as tree cutwidth. The *tree congestion of G* is defined as finding the the minimum $c(G : T)$ by considering every possible tree of G:

$$t(G) = \min_{T_G}\{c(G : T)\}.$$

Similarly, *the spanning tree congestion of* G looks at every spanning tree of G instead of every single tree. That is we consider every possible spanning tree of G and find the minimum $c(G:T)$:

$$s(G) = \min_{S_G}\{c(G:T)\}.$$

For example, for $K_{1,2,2}$ we can create multiple spanning trees. Figure 6 below shows three possible spanning trees for $K_{1,2,2}$.



S_1 S_2 S_3

Figure 6: Three spanning trees of $K_{1,2,2}$

Tables 1, 2, and 3 on the next page list $c(g, S_i)$ for each edge in their corresponding spanning tree above. In $K_{1,2,2}$, we must consider every possible spanning tree for the graph. We are looking at three possible spanning trees and found $s(K_{1,2,2}) \leq 3$. Three is an upper bound for $s(G)$, so S_3 may be the minimal spanning tree for G. It may be possible, however, that G has a spanning tree with even smaller congestion.

The following theorem relates these concepts together in order to provide bounds for $t(G)$ and $s(G)$.

Theorem (Ostrovskii, [8]): For any connected graph, G,
$m_G = t(G) \leq s(G) \leq |E_G| - |V_G| + 2$.

The inequality $t(G) \leq s(G)$ follows from the definitions of t and s. The set of all spanning trees is a (not necessarily a proper) subset of the set of all trees. If a spanning tree provides the minimum $c(G:T)$ for all trees then $t(G) = s(G)$. However, there may be an instance in which the spanning tree is not the minimum. In that case, $t(G) < s(G)$. Recall that with trees, $|V_T| = |V_G|$ and $|E_T| = |V_G| - 1$. For each edge we consider the detour P_g in T. By starting with a graph, G, we can take away

$$|E_G| - |E_T| = |E_G| - (|V_G| - 1) = |E_G| - |V_G| + 1$$

edges to create one spanning tree. For every edge of G that we remove, there is a new detour in the tree so $|E_G| - |V_G| + 1$ detours are created. An edge, g, in T belongs to at most all of the detours therefore edge g belongs to

Edge	**Detours**	**Congestion**
g	$(m_1, n_1), (m_1, n_1, l_1, n_2), (m_1, n_1, l_1), (m_1, n_1, l_1, n_2, l_2)$	4
h	$(m_1, n_1, l_1, n_2), (m_1, n_1, l_1), (m_1, n_1, l_1, n_2, l_2), (n_1, l_1), (n_1, l_1, n_2, l_2)$	5
i	$(m_1, n_1, l_1, n_2), (m_1, n_1, l_1, n_2, l_2), (n_1, l_1, n_2, l_2), (n_2, l_1)$	4
j	$(m_1, n_1, l_1, n_2, l_2), (n_1, l_1, n_2, l_2), (n_2, l_2)$	3

Table 1: Congestion in S_1. $c(K_{1,2,2} : S_1) = 5$

Edge	**Detours**	**Congestion**
g	$(m_1, n_2, l_1, n_1), (m_1, n_2), (m_1, n_2, l_1), (m_1, n_2, l_1, n_1, l_2)$	4
h	$(m_1, n_2, l_1, n_1), (m_1, n_2, l_1, n_1, l_2), (n_1, l_1), (n_2, l_2)$	4
i	$(m_1, n_2, l_1, n_1, l_2), (n_1, l_2)$	3
j	$(m_1, n_2, l_1, n_1), (m_1, n_2, l_1), (m_1, n_2, l_1, n_1, l_2), (n_2, l_1), (n_2, l_1, l_2)$	5

Table 2: Congestion in S_2. $c(K_{1,2,2} : S_2) = 5$

Edge	**Detours**	**Congestion**
g	$(m_1, n_1), (n_1, m_1, l_1), (n_1, m_1, l_2)$	3
h	$(m_1, n_2), (n_2, m_1, l_1), (n_2, m_1, l_2)$	3
i	$(m_1, l_2), (n_1, m_1, l_2), (n_2, m_1, l_2)$	3
j	$(m_1, l_1), (n_1, m_1, l_1), (n_2, m_1, l_1)$	3

Table 3: Congestion in S_3. $c(K_{1,2,2} : S_3) = 3$

at most $|E_T| = |V_G| + 1$ detours. Remember, g is also used as a detour for itself thus the maximum number of detours g is apart of is $|E_T| = |V_G| + 2$ so $s(G) \leq |E_G| - |V_G| + 2$.

Ostrovskii devoted his paper to the minimization of the edge-congestion over all trees with the same vertex set as G and over all spanning trees of G.

2 Background

Graph theory has been useful in applications such as networking, circuit layout, and code design. Cutwidth is one graph theory problem related to these purposes by identifying the optimal way to arrange networks or circuits in such a way that edges are evenly distributed across to minimize the congestion between locations. Chung [2] was the first to work with linear cutwidth in 1988. Since then, she has been followed by others such as Fransisco Rios [9], Matt Johnson [7] and Megan Holben [6] who worked with linear (lcw) and cyclic (ccw) cutwidths.

Rios proved that for any complete graph, K_n, the cutwidths are equal to the following:

$$lcw(K_n) = \begin{cases} \frac{n^2}{4} & n \text{ is even;} \\ \frac{n^2-1}{4} & n \text{ is odd.} \end{cases}$$

$$ccw(K_n) = \begin{cases} \frac{n^2+8}{8} & \frac{n}{2} \text{ is even;} \\ \frac{n^2+4}{8} & \frac{n}{2} \text{ is odd;} \\ \frac{n^2-1}{8} & n \text{ is odd.} \end{cases}$$

In 2003, Johnson worked with the linear cutwidth and created an upper and lower bound for the cyclic cutwidth of complete bipartite graphs. For any complete bipartite graph $K_{m,n}$:

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

$$ccw(K_{m,n}) = \begin{cases} \frac{mn}{4} & \text{if } m \text{ and } n \text{ are both even;} \\ \frac{n^2+4}{8} & \text{if } m = n \text{ are odd.} \end{cases}$$

Holben extended Johnson's results by finding the exact bounds for many different cases. Her results are drawn from Johnson's second theorem.

$$ccw(K_{m,n}) = \begin{cases} \frac{mn}{4} & m,\ n \text{ even;} \\ \frac{mn+j}{4} & m \text{ odd, } j = \frac{n}{m},\ j \text{ even;} \\ \frac{mn+j+2}{4} & m \text{ odd, } j = \frac{n}{m},\ j \text{ odd;} \\ \frac{mn+2}{4} & m \equiv 2(mod4),\ n \text{ odd, } 2n \geq m; \\ \frac{mn+4}{4} & m \equiv 0(mod4),\ n \text{ odd, } 2n \geq m; \\ \frac{mn+l+2}{4} & m \text{ even, } n \text{ odd, } 2n < m,\ l \text{ even,} \\ & m - ln < 2n,\ m - ln \equiv 2(mod4); \\ \frac{mn+l+4}{4} & m \text{ even, } n \text{ odd, } 2n < m,\ l \text{ even,} \\ & m - ln < 2n,\ m - ln \equiv 0(mod4); \\ \frac{mn+l+4}{4} & m \text{ even, } n \text{ odd, } 2n < m,\ l \text{ even and } n \text{ are both even,} \\ & m - ln < 2n,\ m - ln \equiv 3(mod4); \\ \frac{mn+l+4}{4} & m, n \text{ odd, } m < n,\ l \text{ odd,} \\ & n - ln < 2m,\ n - lm \equiv 2(mod4). \end{cases}$$

Heiko Schröder [11], Dwayne Clarke [3], and S. Bezrukov [1] also looked at cyclic cutwidth. Schröder worked with the cyclic cutwidth of a two-dimensional rectangular graph called a mesh, $P_m \times P_n$. Clarke modified the results and found for a graph, G, which is $P_m \times P_n$ mesh where $m \geq n \geq 3$,

$$ccw(G) = \begin{cases} n-1 & m = n \text{ is even;} \\ n & m = n, n+1, \text{ and } n \text{ is odd or;} \\ & m = n, n+1, n+2, \text{ and } n \text{ is even;} \\ n+1 & \text{if otherwise.} \end{cases}$$

Victor Sciortino [12] explored three-dimensional meshes and proved for a graph, G, which is $P_1 \times P_2 \times Pn$ mesh,

$$ccw(G) = \begin{cases} 1 & n = 1; \\ n+1 & 2 \leq n \leq 5; \\ 6 & n \geq 5. \end{cases}$$

Joeseph Chavez and Rolland Trapp [4] have completed the cyclic cutwidth problem for trees and found that if T is a tree, the $lcw(T) = ccw(T)$. In a joint paper with Bezrukov [1], they embedded the n-dimensional cube onto a line and a grid in ways to minimize the cutwidth as well.

M.I. Ostrovskii [8] used trees as the host graphs, but spanning trees in particular. Stephen Hruska [5] has taken Ostrovskii's theorem and investigated the results for all trees, cyclic graphs, C_n, complete graphs, K_n, complete bipartite graphs, $K_{m,n}$, and grids. My paper extends Hruska's research to complete tri-partite graphs. Hruska's results follow:

2.0.1 Preliminaries

$t(G)$ and $s(G)$ of Some Families

- Trees

  $m_G = t(G) = s(G) = 1$

  Suppose G is a tree. Since there is only one edge-disjoint path from one vertex to another, $m_G = t(G) = 1$. In trees, $|E_G| = |V_G| - 1$. Therefore $|E_G| - |V_G| + 2 = (|V_G| - 1) - |V_G| + 2 = 1$. Thus, $1 \leq s(G) \leq 1$ and so $s(G) = 1$.

- Cyclic Graphs

  $m_G = t(G) = s(G) = |E_G| - |V_G| + 2 = 2$

  A cyclic graph, G, is a graph with vertices, V_i, where $i = 1, 2, ..., n-1$ and edges (v_i, v_{i+1}) and (v_n, v_1). Principally, a cyclic graph is an n-gon. There are two edge-disjoint paths from one vertex to an adjacent one, therefore $m_G = t(G) = 2$. $|E_G| = |V_G|$, so $|E_G| - |V_G| + 2 = 2$ and thus $s(G) = 2$.

- Complete Graphs

  $m_G = t(G) = s(G) = n - 1$

  $|E_G| - |V_G| + 2 = \frac{n^2}{2} - \frac{3n}{2} + 2$

  A graph with an edge between every pair of vertices is called complete. Notice that in this case $s(G)$ is not always equal to the upper bound. As graphs become more complicated, m_G, $s(G)$, and the upper bound $|E_G| - |V_G| + 2$ diverge.

- Complete Bipartite Graphs

  $m_G = t(G) = s(G) = 1$ when $m = 1$

  $m_G = t(G) = n$; $s(G) = m + n - 2$ when $m \geq 2$

  $|E_G| - |V_G| + 2 = mn - (m + n) + 2.$

- Grids

G	$m_G = t(G)$	$s(G)$	$\|E_G\| - \|V_G\| + 2$
$P_1 \times P_n$	1	1	1
$P_2 \times P_2$	2	2	2
$P_2 \times P_n (n > 2)$	3	3	n
$P_3 \times P_n$	3	3	$2n - 1$
$P_m \times P_m (4 \leq m)$	4	m	$m^2 - 2m + 2$
$P_m \times P_n (4 \leq m < n$, m odd)	4	m	$mn - (m + n) + 2$
$P_m \times P_n (4 \leq m < n$, m even)	4	$m + 1$	$mn - (m + n) + 2$

3 Main Results

3.1 Complete Tripartite Graphs

Recall that a complete tripartite graph, $K_{m,n,l}$, is a graph that contains three disjoint sets of vertices M, N, and L, with $|M| = m, |N| = n$, and $|L| = l$, such that every vertex in M is joined by an edge to every vertex in N, every vertex in M is joined by an edge to every vertex in L, and every vertex in N is joined by an edge to every vertex in L.

Theorem 1. *For $G = K_{m,n,l}$, with $m \leq n \leq l$:*

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

Proof:

Let G be a complete tripartite graph and let M denote the set of vertices, numbered $1, 2, ..., m$, let N denote the set of vertices numbered $m+1, m+2, ..., m+n$, and let L denoted the set of vertices numbered $m+n+1, m+n+2, ..., m+n+l$, with $m \leq n \leq l$.

- Proof when $m = 0$

  If $m = 0$ then G yields the complete bipartite graph and results hold from Hruska's theorem.

- Proof when $m = 1$

  - **Case 1 Same Set:** $u, v \in N$ **or** $u, v \in L$

    Let u, v be vertices in G such that $u, v \in N$. For each vertex y in L, we can produce one edge-disjoint path of length two, uyv. Since $|L| = l$, this is a total of l paths. Additionally, we produce one more path, uxv, for the one vertex, x, in M. So, the number of edge-disjoint paths in G in this case is $l+1$. Similarly, if $u, v \in L$, the number of edge-disjoint paths in G is $n+1$. Because $n \leq l$, $l+1$ is the maximum number of edge-disjoint paths when $m = 1$ and when u and v are in the same set.

  - **Case 2 Different Sets:** $(u \in M, v \in N)$ **or** $(u \in M, v \in L)$ **or** $(u \in N, v \in L)$

    Let $u \in M, v \in N$. There is one direct path, uv, from u to v. Additionally, for each vertex x in L we produce one more edge-disjoint path of length two, uxv. Since $|L| = l$, this is a total of l paths, thus, the maximum number of edge-disjoint paths in G in this case is also $l+1$. Analogously, when $u \in M, v \in L$ the maximum number of edge-disjoint paths in G is $n+1$. When $u \in N, v \in L$ there are three ways to create edge-disjoint paths. First, there is the direct connection between u and v. Next, there is one path of length two, uxv, for the one vertex in set M. Finally, for each set N and L, choose a distinct vertex $y \in N$ ($y \neq u$) and $z \in L$ ($z \neq v$) and create $(n-1)$ paths of length three, $uzyv$, for every y in set N. Thus, in G and the maximum number of edge-disjoint paths is $2+(n-1) = n+1$. Because $n \leq l$, $n+1 \leq l+1$ thus $l+1$ is the maximum number of edge-disjoint paths when $m = 1$ and when u and v are in different sets.

As a result, no matter what sets u and v are in, the maximum number of edge-disjoint paths is $l+1$.

- Proof when $m \geq 2$
  - **Case 1 Same Set:** $u, v \in M$ **or** $u, v \in N$ **or** $u, v \in L$

    Let u, v be vertices in G such that $u, v \in M$. For every vertex x in set N we create one path of length two, uxv, for a total of n paths and for every vertex y in set L we create 1 path of length two, uyv, for a total of l paths. The maximum number of edge-disjoint paths in G in this case is $n+l$. Using the same methods, when $u, v \in N$ the maximum number of edge-disjoint paths is $m+l$. When $u, v \in L$ the maximum number of edge-disjoint paths is $m+n$. Because $m \leq n \leq l$, the maximum number of edge-disjoint paths when $m \geq 2$ and when u and v are in the same set is $n+l$.
  - **Case 2 Different Sets:** $u \in M, v \in N$ **or** $u \in M, v \in L$ **or** $u \in N, v \in L$

    An example when u and v are in different sets is shown in Figure 7 for $K_{3,4,5}$. Let $u \in M, v \in N$. We have three ways to create edge-disjoint paths in G. First, there is one direct connection between u and v. For each set M and N, choose a distinct vertex $y \in M$ ($y \neq u$) and $z \in N$ ($z \neq v$) and create $(m-1)$ paths of length three, $uzyv$, for every y in set M. We also have one path of length two, uxv, for every vertex x in set L. This produces a total of l paths. Thus, we can create a maximum of $m+l$ edge-disjoint paths in this case. Similarly, when $u \in M, v \in L$ we can create $m+n$ edge-disjoint paths and when $u \in N, v \in L$ we can create $m+n$ edge-disjoint paths. Because $m \leq n \leq l$, the maximum number of edge-disjoint paths when $m \geq 2$ and when u and v are in different sets is $n+l$.

    The maximal number of edge-disjoint paths in Figure 7 is $m+l = 8$.

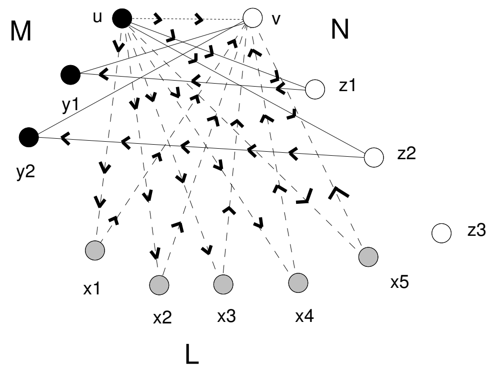


Figure 7: $K_{3,4,5}$ Eight edge-disjoint paths.

As a result, the maximum number of edge disjoint paths connecting any two vertices when $m \geq 2$ is $n + l$.

We conclude that $m_G = t(G) = l + 1$ when $m = 1$ and $m_G = t(G) = n + l$ when $m \geq 2$ which completes the proof of theorem 1.

Theorem 2. *For $G = K_{m,n,l}$ with $m \leq n \leq l$:*

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

Proof

- **Proof when $m \geq 2$:**

  Let G be a graph and let M denote the set of vertices numbered $1, 2, ..., m$, let N denote the set of vertices numbered $m+1, m+2, ..., m+n$, and let L denote the set of vertices numbered $m+n+1, m+n+2, ..., m+n+l$, with $m \leq n \leq l$. At this time we will provide a lower bound for $s(G)$.

  A complete tripartite graph with $m \geq 2$, has no spanning tree that is a star. In any spanning tree, S, of such a graph with $m \geq 2$, there must be a path with length of at least three. Consider a path with length of at least three, P, and denote a subset of the path as $P_1 = (m+1, m+n+1, m+2, m+n+2)$. Figure 8 shows how we build the spanning tree. Because P_1 has length three, we know there must be a parent edge, g, in middle of our path. Let $g = (m+n+1, m+2)$ and notice that g splits the vertices into two distinct regions, $A = \{m+1, m+n+1\}$ and $B = \{m+2, m+n+2\}$. Edge g is used for detours $(m+1, m+n+2)$ and $(m+2, m+n+1)$. Every remaining vertex in set N contributes one to $c(g : S)$ and every remaining vertex in set L contributes one to $c(g : S)$ for a total of two counts to edge g. Without loss of generality, suppose the remaining vertices from set N and set L are in region A. G is a complete tripartite graph, hence no matter what region the remaining vertices from set N and set L are in they will always need to be connected to the vertices in their opposite set in the opposite region. Therefore, we can say $c(g, S) \geq 2 + (n-2) + (l-2)$.

Because $m \geq 2$, we must continue building our spanning tree using vertices from set M. As before, no matter what set the vertices from set M are in, every vertex from set M will always need to be connected to all vertices from set N and set L in the opposite region. Therefore, g will be used in at least two more detours for every vertex in set M. Thus, set M contributes a total of $2m$ to $c(g, S)$. Therefore, in any spanning tree, there is always an edge used in at least $(2m + (n-2) + (l-2) + 2 = (2m + n + l) - 2$ detours, making $c(G : S) \geq (2m + n + l) - 2$.
The congestion of g is true for any spanning tree of G because there will always be a path of length of at least three when $m \geq 2$, therefore, $s(K_{m,n,l}) \geq (2m + n + l) - 2$.

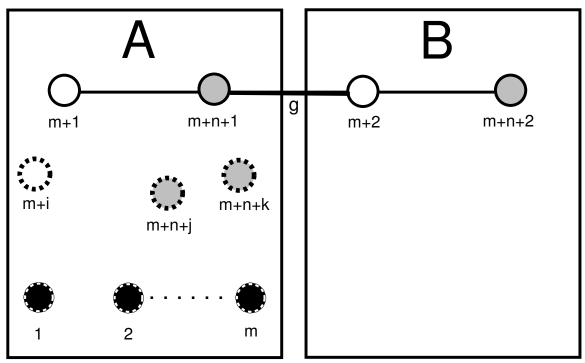


Figure 8: Building the spanning tree.

By creating a minimum spanning tree where $c(G : S) = (2m+n+l) - 2$ we create an upper bound for $s(K_{m,n,l})$ to confirm $s(K_{m,n,l}) = (2m+n+l) - 2$.

Once again, G is a graph and M denotes the set of vertices labeled $1, 2, ..., m$, N denotes the set of vertices labeled $m+1, m+2, ..., m+n$, and L denotes the set of vertices labeled $m+n+1, m+n+2, m+n+l$, with $m \leq n \leq l$. We create the minimum spanning tree, S, as shown in Figure 9 as follows:

Connect all vertices from set M and all vertices from set L to vertex $m+1$ from set N. Attach vertex $(m+i)$ from set N to $(m+n+i-1)$ from set L, for $i = 2, 3, 4, ..., n$. Consider the parent edge $g = (m+1, m+n+1)$. Vertex $(m+2)$ contributes $m + (l-1)$ counts to g through detours $(m+2, m+n+1, m+1, x)$ for every x in set M and in detours $(m+2, m+n+1, m+1, m+n+i)$ where $i \neq 1$. Vertex $(m+n+1)$ contributes $m + (n-1)$ counts to g in detours $(m+n+1, m+1, x)$ for every x in set M and in detours $(m+n+1, m+1, m+n+j, m+i)$ where $i \neq 2$ and $j \neq 1$. Therefore, edge g is used in $[m + (n-1) + m + (l-1)] = (2m + n + l) - 2$

detours concluding, $c(g,S) = (2m+n+l)-2$. By inspection, the remaining parent edges in S have detours equal to that of g.

For every vertex incident to a child edge from set M, h, $c(h,S) = n+l$. Similarly, for every vertex incident to a child edge from set N, r, $c(r,S) = m+l$. Again, for every vertex incident to a child edge from set L, q, $c(q,S) = m+n$. Consequently, since $m+n \leq m+l \leq m+n \leq (2m+n+l)-2$, $c(K_{m,n,l} : S) = (2m+n+l)-2$ and $s(K_{m,n,l}) \leq (2m+n+l)-2$.

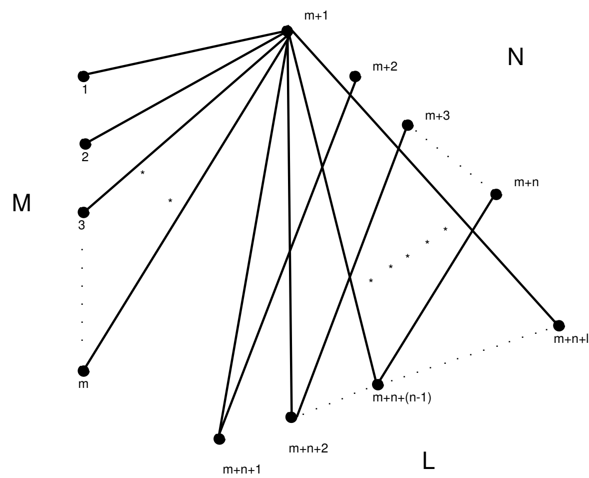


Figure 9: Generic spanning tree that minimizes $c(G : S)$ for $K_{m,n,l}$ when $m \geq 2$

We have provided a lower bound and an upper bound for $s(G)$ when $m \geq 2$, thus for any spanning tree we can confirm $s(G) = (2m+n+l)-2$.

- **Proof when $m = 1$:**

Remember, G is a graph and M denotes the set with one vertex labeled 1. Now let N denote the set of vertices labeled $2,3,...,n+1$ and L denote the set of vertices labeled $n+2, n+3, ..., n+l+1$, with $m \leq n \leq l$. S is one spanning tree of G.

Recall that in trees, $|E_G| = |V_G| - 1$ and in complete tripartite graphs, $m \leq n \leq l$ so when $m = 1$, $n \geq 1$ and $l \geq 1$. We will create a minimum spanning tree where $c(G : S) = (l+1)$ to provide an upper bound for $s(G)$. As shown in Figure 10, produce the minimum spanning tree, S, as follows:

Connect every vertex from set N and every vertex from set L to 1 in set M to generate a spanning tree that is a star. Consider the edge $g = (1, n+2)$. Edge g is used in:

  - one direct connection from 1 to $n+2$.

– each detour from set N to vertex $n+2$ of set L through 1, $(i, 1, n+2)$, for $i = 2, 3, ..., n+1$ in set N. So, all the vertices in set N contribute n counts to g.

Every vertex from sets M and N contribute a total of $n+1$ to the congestion of g. The congestion on all of the edges incident to the vertices in set L is $n+1$. Similarly, all of the edges incident to the vertices in set N is $l+1$.

Figure 10 shows the spanning tree star that minimizes $c(G : S)$ for $K_{m,n,l}$ when $m = 1$. Notice that the longest path has length two. Because $m = 1$ and $n \leq l$, the number of vertices from set L determine how many detours use edge g. Consequently, since $n + 1 \leq l + 1$, $c(G : S) = (l + 1)$ and $s(G) \leq (l + 1)$.

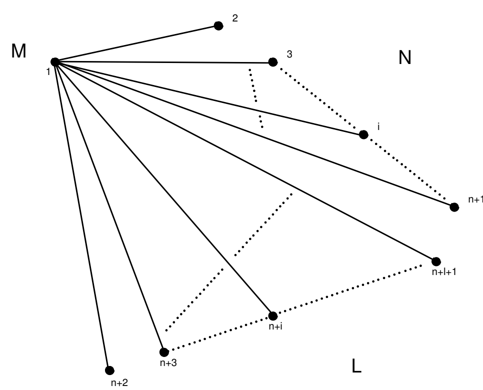


Figure 10: Generic spanning tree that minimizes $c(G : S)$ for $K_{m,n,l}$ when $m = 1$

At this time we will prove the lower bound for $s(K_{m,n,l})$ to show that $s(K_{m,n,l}) = l + 1$. In any spanning tree, S, there is a path of length at least three or all paths are of length two. When all paths are of length two S is a star and $c(G : S) = (l + 1)$. So, now assume S is not a star. Then there must be a path of length at least three. We will refer to the lower bound case when $m \geq 2$ and use the previous notation to label the vertices from set M, N, and L.

Recall that in the last case we let P be a path and denoted a subset of the path as $P_1 = (m + 1, m + n + 1, m + 2, m + n + 2)$. As in this case when $m = 1$, every vertex in set N contributes one to g and every vertex from set L contributes one to g for a total of two counts to $c(g : S)$. No matter what region we add vertices from set N or vertices from set L to they will always need to be connected to the vertices in the opposite set in the opposite region. So, again we have $c(g : S) \geq 2 + (n - 2) + (l - 2)$.

To be a complete tripartite graph, we must build onto our spanning tree using vertex 1 from set M. As the case when adding vertices from the other sets, no matter what region vertex 1 is added to it will always need use g in a detour to be connected to both vertices in sets N and L the opposite region. Therefore, g will be used in two more detours for the one vertex in set M. Thus, vertex 1 from set M contributes 2 to $c(g, S)$. Therefore, in any spanning tree, there is always an edge used in at least $(2 + (n - 2) + (l - 2) + 2(1) = (n + l)$ detours, making $c(G : S) \geq (n + l)$. Obviously, $(l + 1) \leq (n + l)$ thus, $s(G) \geq (l + 1)$.

We have provided an upper bound, $s(G) \leq (l + 1)$, and a lower bound, $s(G) \geq l + 1$, for $s(G)$, thus, for any spanning tree we can confirm that when $m = 1$, $s(G) = (l + 1)$.

We proved that for $G = K_{m,n,l}$ with $m \leq n \leq l$, $s(G) = l + 1$ when $m = 1$ and $s(G) = (2m + n + l) - 2$ when $m \geq 2$.

Theorem 3. *For $G = K_{m,n,l}$ with $m \leq n \leq l$:*

$$|E_G| - |V_G| + 2 = (mn + ml + nl) - (m + n + l) + 2$$

Remember that for any connected graph, Ostrovskii provided bounds for $t(G)$ and $s(G)$. Recall the upper bound is $|E_G| - |V_G| + 2$. Each vertex in set M has degree $m(n + l)$ and each vertex in set N has degree nl thus $|E_G| = m(n + l) = mn + ml + nl$. It is evident that the number of vertices is $m + n + l$. Therefore, $|E_G| - |V_G| + 2 = (mn + ml + nl) - (m + n + l) + 2$.

4 Conclusion

In this paper we have provided the maximum number of edge-disjoint paths, the tree congestion, and spanning tree congestion for complete tri-partite graphs. With the many applications that cutwidth is used in, it would be useful to compare results using spanning trees versus other host graphs. It is interesting to note that using the spanning tree for the complete bi-partite graph, $K_{m,n}$, saves on congestion when compared to the linear cutwidth of $K_{m,n}$. However, in other cases such as grids there are no or only very little savings. Further investigation is still needed for other families of graphs as well such as complete n-partite graphs, three-dimensional grids, and product graphs.

Acknowledgments

I would like to thank Dr. Joseph Chavez and Dr. Rolland Trapp for guiding the *Research Experience for Undergraduates* (REU) program at the California State University, San Bernardino (CSUSB). I would also like to thank the other REU participants for the fun and memorable experiences they brought to the program. The REU Program at CSUSB was made possible through funding through CUSUB and a National Science Foundation grant (NSF-REU Grant DMS-0453605).

References

[1] S.L. Bezrukov, J.D. Chavez, L.H. Harper, M. Röttger, U.-P. Schroeder. The Congestion of n-Cube Layout on a Rectangular Grid, *Discrete Mathematics*. 213 (2000) 13-19.

[2] F.R.K. Chung. Labelings of Graphs, *Selected Topics in Graph Theory*, Vol. 3, Academic Press, San Diego, 1988. 151-168.

[3] D.W. Clarke. The Cyclic Cutwidth of Mesh Cubes, Masters Thesis, Cal State Univ., San Bernardino, 2002.

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

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

[6] M.L. Holben. The Cyclic Cutwidth of Complete Bipartite Graphs, REU Project, Cal State Univ., San Bernardino, 2003.

[7] M. Johnson. The Linear and Cyclic Cutwidth of the Complete Bipartite Graph, REU Project, Cal State Univ., San Bernardino, 2002.

[8] M.I. Ostrovskii. Minimal Congestion Trees, *Discrete Mathematics*. 285 (2004)219-226.

[9] F.R. Rios. Complete Graphs as a First Step Toward Finding the Cyclic Cutwidth of the n-Cube, *Cal State Univ., San Bernardino McNair Scholar's Program Summer Research Journal*, 1996.

[10] J. Rolim, O. Sýkora, I. Vrťo. Optimal Cutwidths and Bisection Widths of 2- and 3-Dimensional Meshes, *Graph-Theoretic Concepts in Computer Science*, Springer, Berlin, 1995. 252-264.

[11] H. Schröder, O. Sýkora, I. Vrťo. Cyclic Cutwidth of the Mesh, *SOFSEM'99: Theory and Practice of Informatics*, Springer, Berlin, 1999. 449-458.

[12] V. Sciortino. The Cyclic Cutwidth of a $P_1 \times P_2 \times Pn$ Mesh, REU Project, Cal State Univ., San Bernardino, 2002.