On Tree Congestion of Graphs

Stephen Hruska

August 20, 2004

Abstract

We investigate a theorem relating detours, tree congestion, and spanning tree congestion of a graph. Specifically, we calculate exact formulas for $t(G)$, $s(G)$, and the upper bound $|E_G| - |V_G| + 2$ for various families of graphs, including grids and complete bipartite graphs.

1 Background

The question of cutwidth has been addressed much in the literature because of its applications to networking and circuit design. Linear cutwidth was first considered by Chung [2] in 1988. Since then, she has been followed by others such as Rios [7] and Johnson [5] who worked with linear and cyclic cutwidths, Clarke [3], Holben [4], and Schröder [9] who also looked at cyclic cutwidth, and Bezrukov [1], who used a grid as the host graph. We follow Ostrovskii [6] in using trees as the host graphs, but spanning trees in particular.

2 Introduction

In this paper, G will denote a connected graph with edge set E_G and vertex set V_G, and T will be a tree such that $V_T = V_G$.

If u and v denote specific vertices in G, then $m(u, v)$ is defined to be the maximal number of edge-disjoint paths in G connecting those vertices. Considering all possible pairs of vertices, we determine the maximum over every $m(u, v)$ and call it m_G – that is,

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

For example, in the complete graph K_4, shown in Figure 1 below, there are a maximum of three edge-disjoint paths between vertices 1 and 3, so $m(1, 3) = 3$. Since there is no vertex of degree greater than three, it is impossible to find four edge-disjoint paths between any pair of vertices, which implies that the maximum number of edge-disjoint paths we will be able to find between any pair of vertices is three, or $m_G = 3$.

When discussing edge congestion, it is common to embed a graph into a line, which simply means that we begin with the same vertices that are in the

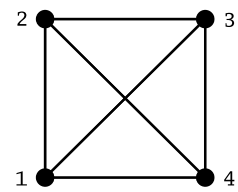

Figure 1: $G = K_4$

original graph, lay them out in a line in any order, and connect them with edges in that order. We then draw edges connecting every pair of vertices that were connected in the original graph. An example of embedding K_4 into a linear host might look like Figure 2, with the dotted lines representing the edges of K_4. After embedding K_4 into the line, we would like to count the number of edges passing between any two vertices (indicated by the vertical lines in Figure 2), find the maximum, and then find the minimum of these maximums over all possible orders of vertices. This minimum is referred to as the linear cutwidth ($lcw(K_4) = 4$).

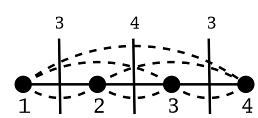

Figure 2: K_4 embedded in a linear host

Tree congestion could essentially be thought of as tree cutwidth. Instead of embedding K_4 into a line, we embed it into a tree, which means that we begin with the same vertices and connect them in such a way that they form a tree. If we are embedding into a spanning tree, then we can only connect vertices that are connected in the original graph. The two possible trees, up to isomorphism, are pictured in Figures 3 and 4.

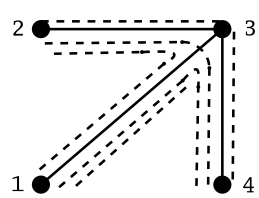

Figure 3: T_1

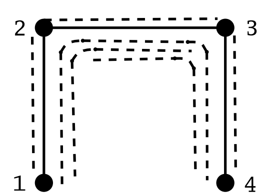

Figure 4: T_2

If we consider an edge g of G, say from 1 to 4, there is a corresponding path in each tree. In T_1 the path is from 1 to 3 to 4, while in T_2 the path is from 1 to 2 to 3 to 4. Each of these paths is called a detour for g, denoted P_g (again, the dotted lines in Figures 3 and 4 represent these detours). Likewise, if g is the edge connecting 2 and 3, then in both trees the detour for g is just the edge itself. A *T-layout L of G* is the collection of all detours in a given tree. In this case,

$$L_{T_1} = \{(1,3,2),(1,3),(1,3,4),(2,3),(2,3,4),(3,4)\}$$

and

$$L_{T_2} = \{(1,2),(1,2,3),(1,2,3,4),(2,3),(2,3,4),(3,4)\}.$$

Of course, $|L| = |E_G|$ for all graphs G and all T-layouts L.

Looking at a specific edge h in T, the *congestion of L in h* is defined as the number of times h appears in L or, equivalently, the number of detours of which h is a part, and it is denoted

$$c(h,L) = |\{P_g \in L : h \in P_g\}|.$$

In the diagrams, this would be the number of dotted lines following along h. If we find the maximum $c(h,L)$ by looking at every edge h of T, we obtain the *congestion of L*, denoted

$$c(L) = \max\{c(h,L) | h \in E_T\}.$$

Continuing the above example, we obtain Tables 1 and 2.

Table 1: Congestions for T_1

h	$c(h, L_{T_1})$
(1,3)	3
(2,3)	3
(3,4)	3

Table 2: Congestions for T_2

h	$c(h, L_{T_2})$
(1,2)	3
(2,3)	4
(3,4)	3

So $c(L_{T_1}) = 3$ and $c(L_{T_2}) = 4$.

Finally, we present the definitions for the *tree congestion of G* and the *spanning tree congestion of G*. The tree congestion of G is the minimum $c(L)$ that we can find if we consider every possible tree:

$$t(G) = \min\{c(L) : \forall \text{ trees } T\}.$$

The spanning tree congestion of G is similarly defined, except we look at all of the spanning trees only, not every single tree:

$$s(G) = \min\{c(L) : \forall \text{ spanning trees } T \text{ of } G\}.$$

Thus, for $G = K_4$, $t(G) = \min\{3,4\} = 3$ and (since both T_1 and T_2 are spanning trees) $s(G) = \min\{3,4\} = 3$.

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

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

Proof.

- The inequality $t(G) \leq s(G)$ follows immediately from the fact that the set of all spanning trees is a (not necessarily proper) subset of the set of all trees. If a spanning tree is the one that provides the minimum $c(L)$ for all trees, then $t(G) = s(G)$, but it is possible that a tree that is not a spanning tree may provide the minimum, in which case $t(G) < s(G)$.

- In obtaining $s(G) \leq |E_G| - |V_G| + 2$, we recall that a tree with $|V_T| = |V_G|$ vertices has $|E_T| = |V_G| - 1$ edges. If we start with the graph G, which has $|E_G|$ edges, we can take away

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

edges to obtain a spanning tree T. For each edge of G that is removed, we create a new detour through the tree, so $|E_G| - |V_G| + 1$ new detours are created. Since an edge h in the tree can only be part of at most all of these new detours, and since an edge in a spanning tree is a detour for itself, the maximum number of detours that any edge can be part of is

$$(|E_G| - |V_G| + 1) + 1 = |E_G| - |V_G| + 2.$$

- The final part of this proof will be broken into two steps: showing $m_G \leq t(G)$ and $m_G \geq t(G)$, which implies the equality.

  1. $\mathbf{m_G \leq t(G)}$Let u, v be the vertices that provide the maximum number of edge-disjoint paths in a graph G; that is, $m(u, v) = m_G$. Let $Q_1, Q_2, ..., Q_{m_G}$ be the m_G edge-disjoint paths in G joining u and v, and let $P = (u = u_1, u_2, ..., u_k = v)$ be the path in a tree T joining u and v. Since T is a tree, every vertex $w \in V_T$ is either on P or is connected through some path to a vertex on P. This means that there is a unique vertex x on P that is a minimum distance from w (whether it is w itself or the first vertex w is connected to). If d_T is the standard graph-theoretic distance, then $x = x(w)$ satisfies

$$d_T(x, w) = \min\{d_T(z, w) | z \in P\}.$$

  If we consider $Q_i = (u = y_1, y_2, ..., y_n = v)$, we can check each vertex y_j in order, starting at $u = y_1$ and ending at $v = y_n$, to see if $x(y_j) = u$. If $x(y_j) = u$, we move on to the next vertex, and if $x(y_j) \neq u$ (we can be sure that there is at least one such vertex since v satisfies $x(v) = v \neq u$), we let e be the edge (y_{j-1}, y_j). P_e, the detour for the edge e, must include the edge $(u = u_1, u_2)$. Because this is true for an arbitrary Q_i, it must be true of all of the Qs, making (u_1, u_2) used in at least m_G detours. Therefore, since the tree we were looking at was arbitrary, $c(L) \geq m_G$ for every tree, so $t(G) = \min\{c(L) : \forall \text{ trees } T\} \geq m_G$.

2. $\mathbf{m_G \geq t(G)}$Let d_v denote the degree of the vertex v, and number the vertices in such a way that $d_{v_1} \geq d_{v_2} \geq d_{v_3} \geq ... \geq d_{v_n}$. Ostrovskii provides a lemma in his paper that says for any graph G and any integer M satisfying $d_{v_1} > M \geq m_G$, there is a tree T with $V_T = V_G$ and $c(L) \leq M$. If we suppose that $d_{v_2} > m_G$, then $M = m_G$ satisfies the inequality and we know there is a tree T that gives $c(L) \leq m_G$, which means $t(G) \leq m_G$. The only case left to consider is when $d_{v_2} \leq m_G$.

   Suppose $d_{v_2} \leq m_G$. We would like to show that $t(G) \leq d_{v_2}$. Pick a vertex from V_G of maximal degree, say v_1. Create a tree T by connecting all other vertices directly to v_1; that is,

$$E_T = \{(v_1, v_i) : i \neq 1\}.$$

   If an edge in G had v_1 as one of its vertices, then the detour in T for that edge is of length 1, since v_1 is still connnected to that other vertex. If an edge in G, say (v_i, v_j), does not have v_1 as one of its vertices, then the detour in T for that edge is of length 2, since all such detours are of the form (v_i, v_1, v_j). Therefore, an edge (v_1, v_i) of T is used as part of a detour for $g \in E_G$ if and only if v_i is one of the vertices of g; thus, each edge (v_1, v_i) in T is used in exactly d_{v_i} detours, where $i \neq 1$. Then the maximum number of detours any edge of T is used in is d_{v_2}, so $c(L) = d_{v_2}$, which implies $t(G) \leq d_{v_2}$. Because of the supposition, we know that $t(G) \leq m_G$. $\square$

3 Preliminary Results

Some families of graphs are easy to categorize in terms of $t(G)$ and $s(G)$.

- All trees satisfy $m_G = t(G) = s(G) = |E_G| - |V_G| + 2 = 1$. Since there is only one (edge-disjoint) path from a given vertex to another one, $m_G = 1$. Also, since trees satisfy $|E_G| = |V_G| - 1$, $|E_G| - |V_G| + 2 = |V_G| - 1 - |V_G| + 2 = 1$. Because $1 \leq s(G) \leq 1$, $s(G) = 1$.

- All cyclic graphs C_n satisfy $m_G = t(G) = s(G) = |E_G| - |V_G| + 2 = 2$. Because G is basically an n-gon, there are two edge-disjoint paths from one vertex to an adjacent one, making $m_G = t(G) = 2$. $|E_G| = |V_G|$, so $|E_G| - |V_G| + 2 = 2$, and $s(G) = 2$ as well.

- All complete graphs K_n satisfy $m_G = t(G) = s(G) = n - 1$ and $|E_G| - |V_G| + 2 = \frac{n^2}{2} - \frac{3n}{2} + 2$. In order to get the maximum number of edge-disjoint paths between vertices u and v, we can count the edge going directly from u to v and then $n - 2$ more paths that go from u to another point and then to v, for a total of $n - 1$ edge-disjoint paths. In complete graphs, all trees are spanning trees, so $t(G) = s(G)$ trivially. Finally, $|V_G| = n$ and

$$|E_G| \quad = \quad (n-1) + (n-2) + ... + (n-n)$$

$$\begin{aligned} &= n \cdot n - \frac{n(n+1)}{2} \\ &= \frac{n^2 - n}{2}. \end{aligned}$$

Hence,

$$\begin{aligned} |E_G| - |V_G| + 2 &= \frac{n^2 - n}{2} - n + 2 \\ &= \frac{n^2}{2} - \frac{3n}{2} + 2. \end{aligned}$$

Note that this is the first example where $s(G)$ is not equal to the upper bound in all cases. Many of the smaller, simpler cases always have equality, but as the graphs become larger and more complicated, the two quantities generally seem to diverge.

- $K_{2,n}$ graphs satisfy $m_G = t(G) = s(G) = |E_G| - |V_G| + 2 = n$. The maximum number of edge-disjoint paths between the two vertices in the left set is n, while the maximum between vertices of the right set is 2 and the maximum number between one vertex from the left and one vertex on the right is 2, so $m_G = n$. $|E_G| = 2n$ and $|V_G| = n+2$, so $|E_G| - |V_G| + 2 = 2n - (n+2) + 2 = n$, and $s(G) = n$.

4 Main Results

4.1 Complete Bipartite Graphs ($G = K_{m,n}$, $m \leq n$)

Let M denote the left set of vertices, numbered $1, 2, ..., m$, and let N denote the right set of vertices, numbered $m+1, m+2, ..., m+n$.

- Case 1: $m = 1$

  In this case, G is a tree, so all quantities are 1 as explained in the Preliminary Results.

- Case 2: $m \geq 2$

  1. $m_G = t(G) = n$

     The maximum number of edge-disjoint paths connecting any two vertices in M is n, obtained by using paths of length two, with each path using a different vertex from N as the middle vertex of the path. Between any vertex from M and any vertex from N, we can only obtain m edge-disjoint paths, one for each vertex in M, and likewise between any two vertices of N. Since $n \geq m$, $m_G = n$.

  2. $s(G) = m + n - 2$

     We know that there is no spanning tree with diameter less than 3 because a tree of diameter 2 would have to connect either two vertices in M or two vertices in N, and such a tree is not a spanning tree. Thus, any spanning tree must have diameter at least 3. Since this is true, there must be a path in the tree with length at least 3. We consider such a path $P = (m_1, n_1, m_2, n_2)$ (where $m_i \in M$ and $n_j \in N$) and we denote its middle edge (n_1, m_2) by g. Edge g is used in 2 detours so far, namely the detours for (m_1, n_2) and (n_1, m_2). Written in terms of the number of vertices from M and N, we have $2 = 2 + 2 - 2$. Notice that g effectively splits the four vertices into 2 distinct sets $L = \{m_1, n_1\}$ and $R = \{m_2, n_2\}$ such that there is a vertex from M in each set and a vertex from N in each set. If we build the spanning tree from this path by adding a vertex v (m_3 or n_3) and connecting it with an edge to one of the 2 sets (say L without loss of generality), we see that g must be used in at least one more detour (since there is a vertex in R that was connected to v in G). Therefore, for each vertex we add, we add at least one more detour to the number that g is part of. If we add p vertices from M (so $m = 2 + p$) and q vertices from N (so $n = 2 + q$), we conclude that g must be part of at least

$$\begin{aligned} 2 + 2 - 2 + p + q &= (2 + p) + (2 + q) - 2 \\ &= m + n - 2 \end{aligned}$$

detours. Thus, in any spanning tree, there is always an edge that is used in at least $m + n - 2$ detours, which makes $s(G) \geq m + n - 2$.

Because of the above inequality, if we can create a spanning tree where we always have $c(L) = m + n - 2$, then we will have proved the equality desired. We conjecture that the spanning tree where one vertex of M is connected to each vertex of N and the rest of the vertices of M are each connected to a different vertex of N (see Figure 5) is a spanning tree that always gives $c(L) = m + n - 2$. We consider the tree redrawn as in Figure 6. An edge $g = (1, m + 1)$ is used:

- once for itself;
- once for each of the $n - 1$ detours $(2, m + 1, 1, n_i)$, where $n_i \in N$ and $n_i \neq m + 1$; and
- once for each of the $m-2$ detours $(m+1, 1, n_i, m_j)$, where $n_i \in N$, $n_i \neq m + 1$, $m_j \in M$, and $m_j \neq 1$ or 2.

No other edge is used in more detours, so we see that forming the spanning tree in this way gives $1 + (n - 1) + (m - 2) = m + n - 2$ as the maximum number of detours any edge is part of (which is $c(L)$). Thus we have the desired equality.

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

   Each vertex in M has degree n, so $|E_G| = mn$ (since no two vertices of M are connected to each other. Clearly, the number of vertices is $m + n$. The equality follows immediately.

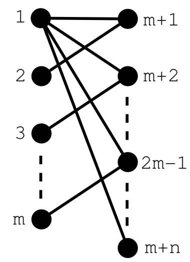

Figure 5: Spanning tree that minimizes $c(L)$ for $K_{m,n}$

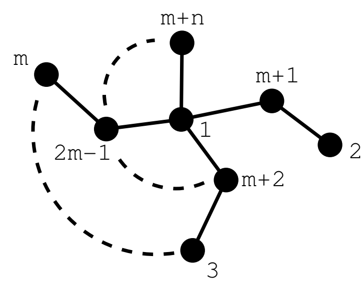

Figure 6: Redrawn spanning tree

4.2 Grids ($G = P_m \times P_n$, $m \leq n$)

When dealing with grids, it is convenient to label each vertex with a pair of coordinates. We let the horizontal numbering range from 1 on the left to n on the right and the vertical numbering range from 1 at the top to m at the bottom (see Figure 7). In addition, it will be useful to consider a sort of dual

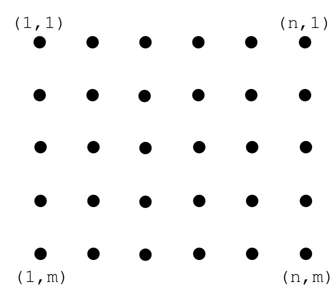

Figure 7: Grid Coordinate System

grid D that will be based on the spanning trees that we choose. We begin with the original grid G, and we place a dual vertex in each region, with one vertex Ω used for the entire outer region. In creating the spanning tree, each edge that is removed is replaced with an edge connecting the two dual vertices it had originally separated. A sample grid G, spanning tree T, and dual grid D are shown in Figure 8. We would also like to have coordinates for the vertices of D, so we will label them in the same way as we did for G, such that the upper-left vertex is $(1, 1)$ and the lower-right vertex is $(n - 1, m - 1)$. Notice that this means each box in G is essentially associated with its upper-left vertex.

Based on this dual grid, we define an *open path in T* as a path in D from one dual vertex to another, but Ω may only be used as an endpoint of this path, if it is used at all. It is clear that every dual vertex must be part of some open

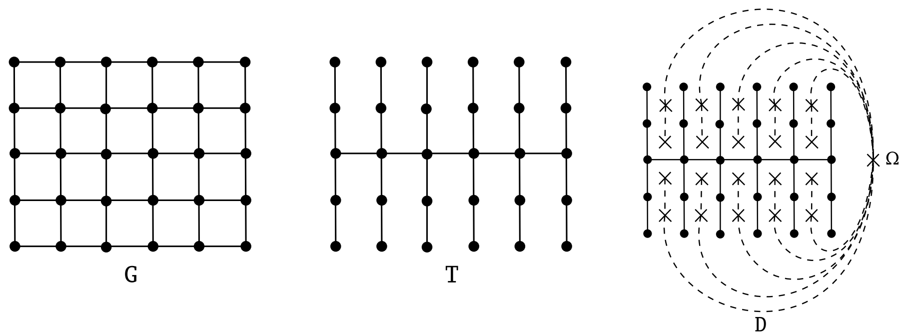

Figure 8: Dual Grid Example

path that connects to Ω, and D must be a tree (otherwise, either T would have a loop or T would be disconnected).

We also define a *dividing path in T* as a path that has both of its end vertices on the boundary of the grid, and we define a *dividing edge* as any edge that is part of a dividing path. Equivalently, an interior edge of T is part of a dividing path if and only if there does not exist an open path from the dual vertex on one side of the edge to the dual vertex on the other side. Trivially, any boundary edge is a dividing path.

Finally, if g is a dividing edge, we consider the open paths P_1 and P_2 from the dual vertices on both sides of g to Ω. Suppose two vertices are adjacent in G (connected by edge h_1) and one of the open paths for g passes between them, but g is not in the detour for h_1. Then if we are following the open path, as soon as we cross between those two vertices, we are blocked by the real detour from getting to Ω, so this open path could not be P_1 or P_2. Therefore, every edge in these two open paths must correspond to a different detour of which g must be a part. Now suppose instead that g is part of a detour for an edge h_2 of G. Then there must be a path in T connecting one vertex of g to one vertex of h_2 and a path connecting the other two vertices such that the two paths do not cross. Since there would be no other open path to get around these paths without going between the vertices of h_2, we know that any detour of which g is a part must correspond to a different edge in P_1 or P_2. Therefore, keeping in mind that we ignored the fact that g is a detour for itself, the exact number of detours of which g is a part is exactly $l(P_1) + l(P_2) + 1$.

With these definitions, we now prove the quantities for $m_G = t(G)$, $s(G)$, and $|E_G| - |V_G| + 2$ for all grids.

- Case 1: $m = 1$

  Again, in this case, G is a tree. All quantities are trivially 1.

- Case 2: $m = n = 2$

  G is C_4, so all quantities are trivially 2.

- Case 3: $2 = m < n$

  1. $m_G = t(G) = 3$

     No vertex has degree greater than 3, so $m_G \leq 3$. The vertices of any inner vertical edge have 3 edge-disjoint paths from one to the other (the vertical edge itself, a path to the left, and a path to the right), so $m_G \geq 3$. Therefore, $m_G = 3$.

  2. $s(G) = 3$

     If we remove all of the horizontal edges on the top row, we create a spanning tree where all of the horizontal edges that are left are used in 2 detours, the outer vertical edges are used in 2 detours, and the inner vertical edges are used in 3 detours. Because we have a lower bound of 3 from m_G, we know $s(G) = 3$.

  3. $|E_G| - |V_G| + 2 = n$

     There are $n - 1$ horizontal edges in each row and n vertical edges, so $|E_G| = 2(n-1) + n = 3n - 2$. Also, there are $2n$ vertices, so $|E_G| - |V_G| + 2 = (3n-2) - (2n) + 2 = n$.

- Case 4: $m = 3$

  1. $m_G = t(G) = 3$

     Even though there are vertices of degree 4, they all lie in the middle horizontal line. Since there are only 3 horizontal edges to get from one side of those edges to the other side, there can only be at most 3 edge-disjoint paths between the vertices of degree 4. If we consider any pair of adjacent vertices such that at least one of them is an inner vertex, we can easily find 3 edge-disjoint paths between them.

  2. $s(G) = 3$

     If we remove all of the horizontal edges on the top row and on the bottom row, we leave a spanning tree in which the horizontal edges are used in 3 detours, the inner vertical edges are used in 3 detours, and the outer vertical edges are used in 2 detours. With a lower bound of 3, we know $s(G) = 3$.

  3. $|E_G| - |V_G| + 2 = 2n - 1$

     $|E_G| = 3(n-1) + 2n = 5n - 3$. $|V_G| = 3n$. Therefore, $|E_G| - |V_G| + 2 = (5n-3) - (3n) + 2 = 2n - 1$.

- Case 5: $4 \leq m = n$

  1. $m_G = t(G) = 4$

     Pick two adjacent inner vertices. There are 4 edge-disjoint paths between them, and there cannot be more than 4 because no vertex has degree greater than 4.

2. $s(G) = m$

In any given spanning tree T, there exists a dividing path P from $(1,1)$ to (m,m). At least one of the vertices on the path must lie on the diagonal from $(1,m)$ to $(m,1)$. If we follow P from $(1,1)$ to (m,m), there is a first such vertex, say (p,q). Every vertex on the diagonal is of the form (r,s) such that $r+s = m+1$, so $p+q = m+1$ as well.

Consider the edge g on P previous to (p,q) and assume without loss of generality that it is horizontal (since we could reflect the grid over the diagonal from $(1,1)$ to (m,m) if it were vertical). Then g is $[(p-1,q),(p,q)]$ There exist open paths from both dual vertices on either side of g to Ω.

Suppose the open paths go to opposite sides of the grid. Then the sum of the lengths of the two open paths must be $m-1$, and since g is a detour for itself, g must be used in at least m detours.

Suppose instead that they go to adjacent sides of the grid. Because P effectively splits the boundary of the grid into two sets (top and right in one set and bottom and left in the other set), the open path above g must go to the top and the one below must go left, or the open path above must go right and the one below must go down. In the former case, the open path above is at least $q-1$ edges long and the open path below is at least $p-1$ long, so (including g itself) g is used in at least

$$\begin{aligned}(q-1)+(p-1)+1 &= p+q-1 \\ &= m+1-1 \\ &= m\end{aligned}$$

detours. In the latter case, the open path above is at least $m-p+1$ long and the open path below is at least $m-q$ long, so g is used in at least

$$\begin{aligned}(m-p+1)+(m-q)+1 &= 2m-(p+q)+2 \\ &= 2m-(m+1)+2 \\ &= m+1\end{aligned}$$

detours. Therefore, no matter what the spanning tree looks like, we know that there is an edge that must be used in at least m detours, so $s(G) \geq m$.

We can always create a spanning tree that gives $c(L) = m$, so $s(G) = m$. If m is odd, we create this tree by removing all horizontal edges from G except the middle row and leaving all of the vertical edges intact (in Figure 9, the bold edges are the ones used in m detours). If m is even, the tree is formed by removing all horizontal edges except the middle two rows and leaving all vertical edges outside the middle rows and only one of the middle edges between the rows (Figure 10).

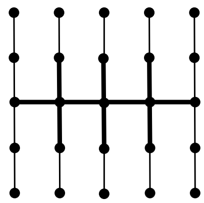

Figure 9: m odd

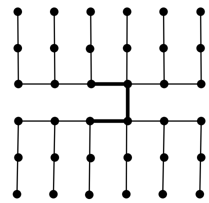

Figure 10: m even

3. $|E_G| - |V_G| + 2 = m^2 - 2m + 2$

$$\begin{aligned}|E_G| &= m(m-1) + m(m-1) \\ &= 2m^2 - 2m,\end{aligned}$$

and $|V_G| = m^2$, so

$$\begin{aligned}|E_G| - |V_G| + 2 &= (2m^2 - 2m) - m^2 + 2 \\ &= m^2 - 2m + 2.\end{aligned}$$

- Case 6: $4 \leq m < n$, m odd

1. $m_G = t(G) = 4$
   See Case 5.
2. $s(G) = m$
   Because of the proof for Case 5, we cannot obtain a lower $s(G)$ by adding more columns of vertices. Also, since the same method for creating the spanning tree still gives $c(L) = m$, we have $s(G) = m$.
3. $|E_G| - |V_G| + 2 = mn - (m+n) + 2$

$$\begin{aligned}|E_G| &= m(n-1) + n(m-1) \\ &= 2mn - (m+n),\end{aligned}$$

and $|V_G| = mn$, so

$$\begin{aligned}|E_G| - |V_G| + 2 &= [2mn - (m+n)] - mn + 2 \\ &= mn - (m+n) + 2.\end{aligned}$$

- Case 7: $4 \leq m < n$, m even

1. $m_G = t(G) = 4$
   See Case 5.

2. $s(G) = m + 1$

Unfortunately, the method in Case 5 for creating a spanning tree with $c(L) = m$, m even, causes the central vertical edge to be used in at least $m + 1$ detours if $m < n$, so we need to prove that is the lower bound and come up with a method that will always give $c(L) = m + 1$.

Consider the section of the grid with $(1, 1)$ and $(m + 1, m)$ as the corners. In any spanning tree T, there is a dividing path P_1 connecting these two vertices, and a second dividing path P_2 connecting $(1, m)$ and $(m+1, 1)$. Assign an orientation to the paths so that the positive direction for P_1 is from $(1, 1)$ to $(m + 1, m)$ and the positive direction for P_2 is from $(1, m)$ to $(m + 1, 1)$. P_1 and P_2 can only intersect one time, either sharing some number of edges or crossing so that a vertex of degree 4 is formed. Also, P_1 and P_2 could be thought of as dividing the boundary of the grid into four sets: top, left, right, and bottom.

If P_1 and P_2 share a vertical edge g anywhere, then T looks essentially like one of the three trees in Figure 11. When this happens,

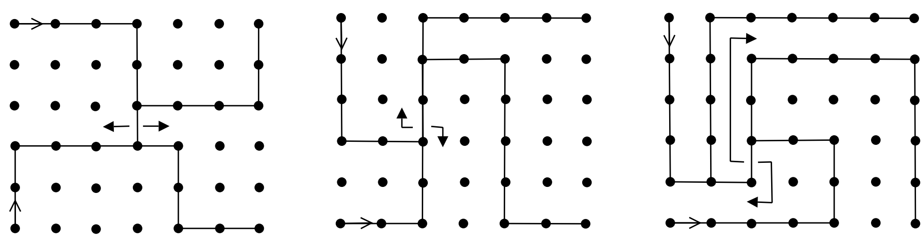

Figure 11: P_1, P_2 share a vertical edge

from the dual vertices on both sides of g there is either an open path crossing the left boundary and an open path crossing the right boundary or there is an open path crossing the top and one crossing the bottom. When they go left and right, the lengths of the open paths sum to at least m and the total number of detours g is part of is at least $m + 1$. When they go up and down, the lengths of the open paths still sum to at least m, and g is still part of at least $m+1$ detours.

We suppose, then, that P_1 and P_2 do not share a vertical edge. Because m is even, there is a middle row of vertical edges. We assume without loss of generality that the intersection of the dividing paths occurs completely above this middle row. If we follow P_1 and P_2 in their positive directions away from the intersection, the next vertical edge g_1 on the middle row must be part of P_1, oriented down, so there is an open path that goes from the dual vertex on the right side of this edge to the right boundary. Likewise, if we follow the paths in the opposite direction, the next edge g_2 on the middle row is part of P_2, oriented up, so there is an open path that goes from the dual vertex on the left side of this edge to the left boundary.

Consider these two dividing edges. If there is no other dividing edge between them, then there is an open path from the dual vertex on the right of g_2 to the dual vertex on the left of g_1 and there is an open path from each to Ω crossing the bottom of the boundary. We would like to minimize the maximum number of detours these edges are part of, and this will happen when the open paths cross the boundary as close to the middle as possible. Since $m+1$ is odd, we suppose that the missing edge is

$$\left[\left(\frac{m}{2}+1, m\right), \left(\frac{m}{2}+2, m\right)\right].$$

Then the open paths from the dual vertices to the left and right of g_2 use at least $\frac{m}{2}$ horizontal edges (including the edge crossing the left boundary) and at least $\frac{m}{2}$ vertical edges (including the edge crossing the bottom). Combining these with g_2 itself, the number of detours g_2 is part of is at least $\frac{m}{2}+\frac{m}{2}+1 = m+1$. By making this number smaller, we would only increase the number of detours g_1 must be part of to at least this many.

If there is another dividing edge between g_1 and g_2 on the middle row that is not part of P_1 or P_2, it must be part of a different dividing path. Then on either side of this dividing edge there must be an open path from the left dual vertex to Ω crossing the bottom of the boundary and another open path from the right dual vertex to Ω crossing the bottom. These two paths are each $\frac{m}{2}$ long, so this other dividing edge must be used in at least $2 \cdot \frac{m}{2} + 1 = m+1$ detours.

Finally, if there is another dividing edge between g_1 and g_2 that is part of P_1 or P_2, then we have a situation similar to the one where there is no dividing edge between. In this case, the dividing path has curved around some, but that only forces g_1 or g_2 to be used in even more detours.

Therefore, $s(G) \geq m+1$. Because the spanning tree formed by removing all horizontal edges except for those in one of the middle two rows and leaving all vertical edges intact always has $c(L) = m+1$ no matter what n is (Figure 12), we have $s(G) = m+1$.

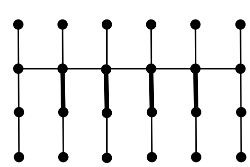

Figure 12: Spanning Tree for $P_m \times P_n$, $m < n$, m even

3. $|E_G| - |V_G| + 2 = mn - (m+n) + 2$
   See Case 6.

5 Conclusion

Table 3 below details all of the results proved in this paper for the various families of graphs. Because of the applications of cutwidth, it would be useful to compare these results with those obtained using other host graphs. In some cases, using a spanning tree as the host graph can provide great savings in terms of congestion. For example, the linear cutwidth of a complete bipartite graph $K_{m,n}$ is either $\frac{mn}{2}$ or $\frac{mn+1}{2}$ (depending on whether mn is even or odd), while $s(K_{m,n}) = m + n - 2$. However, in other cases such as grids, there are no or only very little savings. Rolim [8] shows that for $2 \leq m \leq n$,

$$lcw(P_m \times P_n) = \begin{cases} 2 & \text{if } m = n = 2 \\ m+1 & \text{otherwise} \end{cases},$$

which is very close to the numbers for $s(G)$.

Table 3: Summary of Results

G	$m_G = t(G)$	$s(G)$	$\lvert E_G\rvert - \lvert V_G\rvert + 2$
Trees	1	1	1
C_n	2	2	2
K_n	$n-1$	$n-1$	$\frac{n^2}{2} - \frac{3n}{2} + 2$
$K_{m,n}(2 \leq m \leq n)$	n	$m+n-2$	$mn-(m+n)+2$
$P_1 \times P_n$	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	$m^2 - 2m + 2$
$P_m \times P_n (4 \leq m < n,\ m \text{ odd})$	4	m	$mn-(m+n)+2$
$P_m \times P_n (4 \leq m < n,\ m \text{ even})$	4	$m+1$	$mn-(m+n)+2$

Possibilities for further work with tree congestion and spanning tree congestion include determining what these quantities would be for other families like n-cubes, complete and general n-partite graphs, and cylindrical meshes. For the first and last of these, we conjecture that $s(Q_n) = 2^{n-1}$ and $s(C_m \times P_n) = \min\{2m+2, 2n\}$. One might also consider the bound $|E_G| - |V_G| + 2$ and when it is equal to $s(G)$ (so far, it seems that if these two quantities are equal, then $t(G) = s(G)$).

Acknowledgements

The author would like to thank Dr. Joseph Chavez and Dr. Rolland Trapp for leading the REU at California State University, San Bernardino, as well as the rest of the participants for their input and support. This research was done at the 2004 REU in Mathematics, jointly sponsored by CSUSB and NSF-REU Grant DMS-0139426.

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

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

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

[7] 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.

[8] 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.

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