Math 443/543 Graph Theory Notes 4: Connector Problems

David Glickenstein

September 15, 2008

1 Trees and the Minimal Connector Problem

Here is the problem: Suppose we have a collection of cities which we want to connect by a system of railway lines. Also suppose we know the cost of constructing lines between pairs of cities. How do we construct the system so that the cost is minimal, regardless of the inconvenience to the passengers.

We are trying to construct a network of minimal cost. Here are some observations:

- The solution must be connected.
- The solution should contain no cycles, for if there were any cycles, we could remove one of the edges to get a smaller cost, and passengers could still get from one place to the other. Thus we want every edge to be a bridge.

Such a graph is called a tree.

Definition 1 *A graph with no cycles is called a* forest*. A connected graph with no cycles is called a* tree*.*

Note that each component of a forest is a tree. Trees kind of look like branches of a tree, which is where the name comes from.

Here are some properties of trees.

Theorem 2 *Let u and v be two vertices of a tree G. Then there is exactly one uv-path in G.*

Proof. Since G is connected, there is at least one uv-path. Suppose there were two uv-paths, $P = v_0, v_1, v_2, \ldots, v_k$ and $P' = v'_0, v'_1, v'_2, \ldots, v'_{k'}$ (note that k need not equal k'). If $P \neq P'$, then we can define s and f as

$$s = \min\left\{i : v_{i+1} \neq v'_{i+1}\right\}$$
$$f = \max\left\{i : v_{i-1} \neq v'_{i-1}\right\}.$$

We can then construct two disjoint paths between v_s and v_f, producing a cycle. But there are no cycles in G, so we must have $P = P'$. ■

Theorem 3 *If a (p, q)-graph G is a tree, then $q = p - 1$.*

Proof. We prove this by induction, stating it as "every tree of order p has size $p - 1$." Certainly a tree of order 1 has no edges, and this gives the base case. Now suppose it is true for all trees of order p or less. We need to show that a tree of order $p + 1$ has size p. Since every edge is a bridge, we may remove an edge e to produce two trees, each with order less than $p+1$. If the two trees are T and T' with orders s and s', we know that $s + s' = p + 1$ and at the sizes of T and T' are $s - 1$ and $s' - 1$ by the inductive hypothesis. Thus the size of G is equal to $1 + (s - 1) + (s' - 1) = p$. ■

We come back to the minimal connector problem.

Definition 4 *If G is a connected graph, a subgraph T which is a tree and contains every vertex of G is called a* spanning tree.

We wish to find a spanning tree of minimal value. Let's find a particular tree called an *economy tree*. We construct the subgraph T_E starting with all of the vertices of G and then add edges one at a time:

1. First add the edge of minimal weight.
2. Continue to add edges of minimal weight unless they would form a cycle with the previously added edges.
3. Stop when the graph is connected.

Since no cycles are produced and the graph ends up connected and reaching every vertex, we produce a spanning tree. One might ask whether we can, in fact do this. Note that if the subgraph T_E at some stage is not connected, then consider a path between two different components (which exists since G was connected). If all edges on the path not already in T_E (there must be some) create a cycle when added, then the two components were already connected (by the other part of the cycle), a contradiction.

Note that the economy tree is not necessarily unique. We may have many of them.

The theorem is that the economy tree solves the minimal connector problem.

Theorem 5 *Let (G, ϕ) be a network. The economy tree T_E has minimal weight among all spanning trees.*

Proof. Let T_0 be a spanning tree of minimal weight. We will show that $\phi(T_E) \leq \phi(T_0)$, which implies equality since $\phi(T_0)$ is minimal among all spanning trees. We can order the edges in T_E by the weights, i.e., $E(T_E) = \{e_1, e_2, \ldots, e_{p-1}\}$ where

$$\phi(e_i) \leq \phi(e_{i+1}).$$

Now let e_j be the first edge in T_E which is not in T_0. Let $G_0 = T_0 + e_j$. Since T_0 is a spanning tree, G_0 must have a cycle C. Since T_E is a tree, there must

be an edge e_0 in C which is not in T_E. In particular, $e_0 \in T_0$. We can consider the graph $T_0' = G_0 - e_0$, which has no cycles and is connected, so it is also a spanning tree. We notice that

$$\phi(T_0') = \phi(T_0) + \phi(e_j) - \phi(e_0).$$

Since T_0 is minimal, we have

$$\phi(T_0) \leq \phi(T_0') = \phi(T_0) + \phi(e_j) - \phi(e_0)$$

and hence

$$\phi(e_0) \leq \phi(e_j).$$

However, $\phi(e_j)$ was chosen to have minimal weight among edges in G not in T_E in the construction, which means we must have that

$$\phi(e_0) = \phi(e_j).$$

Thus

$$\phi(T_0') = \phi(T_0).$$

So T_0' is also minimal. We now consider T_0' instead of T_0. We may do the same construction as before, finding the first edge $e_{j'}$ which is in T_E but not T_0'. We note that $j' \geq j+1$. We may continue to do this until we construct a minimal spanning tree which is equal to T_E. ■

See Example in Figure 4.11 in C.

Note, the algorithm of constructing the economy tree is called Kruskal's algorithm.