**The Dynamic Steiner Tree Algorithm**

By: Thomas Wheeler, Paul Nathan, and Weir King

**Introduction**

This report analyzes the modifications made to the current Steiner tree by the dynamic Steiner tree algorithm for different vertex removals. The changes to the tree and the subsequent altering done by the Steiner tree algorithm are graphically displayed.

*What is a Steiner tree?*

Let graph G consist of vertices V, edges E and additionally terminals T ⊂V.

A Steiner tree for G is a subgraph of G with minimum edge weight that contains all T.

One real world example of the usage of a Steiner tree is minimum cost transportation or communication networks. For a communication network T is the set of endpoints that must be connected, V is the set of potential routers, and E is the set of wired connections between routers and endpoints. The Steiner tree for this graph will require the minimum amount of wired connections to fully connect all the endpoints.

*What is the dynamic Steiner tree problem?*

1. Let graph G be as above and S be the Steiner tree for G.
2. A modification step is performed on G:
   1. Add or remove a vertex to V

**OR**

* 1. Add or remove a terminal from T

1. Compute the new Steiner tree S` for the new modified graph G`

The same real world example of a communication network can again be used to represent the dynamic Steiner tree problem. In an already established communications network endpoints are occasionally added or removed; routers can fail and new routers can be added. A dynamic Steiner tree algorithm can be used to find the new Steiner tree for the new communications network.

**Hypothesis**

The alterations made to the Steiner tree by the Steiner tree algorithm will not be visually significant unless certain key vertices are added or removed by modifications.

**Related Work:**

The dynamic Steiner tree problem was first proposed in 1991 by Imase and Waxman. They designed the dynamic algorithm we have implemented and an additional algorithm that does not re-route the existing Steiner tree for each modification. Imaxe and Waxman however do not visualize the real-time changes made to the Steiner tree after each step.

**Approach:**

The dynamic Steiner tree algorithm (hereafter DSTA) is implemented in Python3 with numPY. DSTA starts with a base node and a Steiner tree consisting of just that node. Changes are then made in the form of change requests. The DSTA updates the Steiner tree after every change request and displays the new Steiner tree overlayed on the current graph to the screen.