

Variations on the Euclidean Steiner Tree Problem and Algorithms

Jack Holby
St. Lawrence University

Follow this and additional works at: <https://scholar.rose-hulman.edu/rhumj>

Recommended Citation

Holby, Jack (2017) "Variations on the Euclidean Steiner Tree Problem and Algorithms," *Rose-Hulman Undergraduate Mathematics Journal*: Vol. 18 : Iss. 1 , Article 7.
Available at: <https://scholar.rose-hulman.edu/rhumj/vol18/iss1/7>

ROSE-
HULMAN
UNDERGRADUATE
MATHEMATICS
JOURNAL

VARIATIONS ON THE EUCLIDEAN
STEINER TREE PROBLEM AND
ALGORITHMS

Jack Holby^a

VOLUME 18, No. 1, SPRING 2017

Sponsored by

Rose-Hulman Institute of Technology
Department of Mathematics
Terre Haute, IN 47803
mathjournal@rose-hulman.edu
scholar.rose-hulman.edu/rhumj

^aSt. Lawrence University

ROSE-HULMAN UNDERGRADUATE MATHEMATICS JOURNAL

VOLUME 18, NO. 1, SPRING 2017

VARIATIONS ON THE EUCLIDEAN STEINER TREE PROBLEM AND ALGORITHMS

Jack Holby

Abstract. The Euclidean Steiner Tree Problem (ESTP) involves creating a minimal spanning network of a set of points by allowing the introduction of new points, called Steiner points. This paper discusses a variation on this classic problem by introducing a single Steiner line whose weight is not counted in the resulting network, in addition to the Steiner points. For small sets, we arrive at a complete geometric solution. We discuss heuristic algorithms for solving this variation on larger sets. We believe that, in general, this problem is NP-hard.

Acknowledgements: I would like to thank my adviser, Dr. Sam Vandervelde, for his unparalleled guidance during this research. His ability to ask the right questions, enthusiasm, and willingness to engage in a variety of ideas pushed me greatly. This research is in many ways a combination of our efforts. I'd also like to thank Dr. Patti Frazer-Lock who introduced me to graph theory and purposefully directed me to study Steiner trees.

1 Introduction

Imagine that you are tasked with creating a road network to connect a group of four cities, but you have to use the least amount of road possible. You could easily find a minimum spanning tree (**MST**) among the four cities using Kruskal's algorithm. This resulting minimal spanning tree would in fact be the best solution given only these fixed four cities, or vertices. If we allow the introduction of additional vertices, however, we can do better. Let's look at this very case in Figures 1(a), (b), and (c). We are trying to connect the four cities, A , B , C , and D , using the least amount of road. For the sake of simplicity, we'll assume that the length, or **weight**, of each edge is 1.

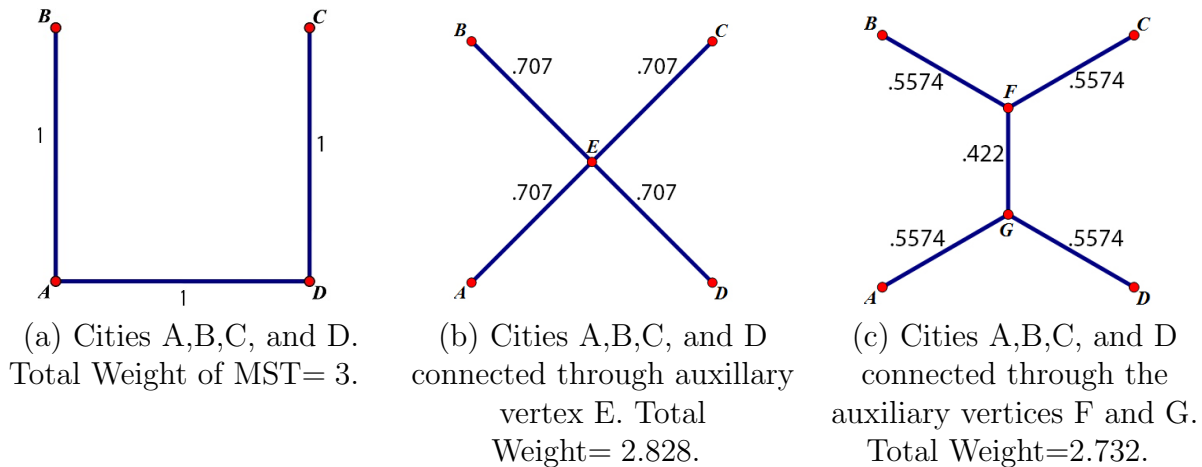


Figure 1: Figures 1(a), (b), and (c) illustrate possible ways to create a network between four cities, or vertices, while allowing for the introduction of additional vertices.

Using direct connections between vertices A , B , C , and D , an MST has weight 3, as shown in Figure 1(a). If we add a new vertex, E , however, and allow connections between E and the original four vertices, we can find an MST of weight 2.828, as illustrated in Figure 1(b). By adding two vertices F and G , as shown in Figure 1(c), we can find a tree with weight 2.732.

In this case, the addition of auxiliary vertices allows us to create a minimum spanning tree of less weight. In general, this is always the case. This observation gave rise to the Euclidean Steiner Tree Problem (ESTP) whose objective has been summed up concisely by Van Laarhoven:

“The objective of the Euclidean Steiner tree problem (ESTP) is to determine the minimal length tree (with respect to the euclidean metric) spanning a set of terminal points, X , while permitting the introduction of extra points (composing a set S of Steiner points) into the network to reduce its overall length [7].”

The Euclidean Steiner Tree Problem has been of great interest to mathematicians because it allows us to create networks of minimum weight to connect a set of points. When applied

to real world problems, the solution can have significant financial implications. We can drastically reduce the amount of road necessary to connect cities or cable needed to connect networks. Because of its financial impact, the ESTP has been well researched and can be solved for relatively small sets of vertices ($n < 2000$).

In this paper, we consider variations on the Euclidean Steiner Tree Problem. Specifically, we include an additional line in the classic problem and attempt to form a Steiner minimal network using this line. This suggests two problems.

1. First, what if we are trying to connect a terminal set of vertices to a line to to create a tree of minimal weight? In this scenario, imagine that you are trying to connect a group of towns to a highway. What is the best way to do this?
2. Second, how should the resulting network look if we can place this line anywhere we choose? We are considering the addition of a “Steiner line” in addition to Steiner points whose weight is not counted in the resulting network. Here, imagine that a number of cities in a county wish to create a highway to connect them, paid for by the federal government, but the county will have to pay for all the roads to connect to the highway. Where is the best placement of that highway?

In Section 2 we review the origins of the Euclidean Steiner Tree Problem, which dates back over 300 years. In Section 3 we introduce the defining characteristics of Steiner trees, which guide the proofs, algorithms, and solutions found in later sections. In Section 4 we present elementary and foundational algorithms on which the search for solutions to the ESTP is predicated. In Section 5 we explore the first variation on the ESTP. In Section 6 we discuss the second variation on the ESTP. In Section 7 we introduce two original algorithms: the first, the Steiner Correction Algorithms, provides a complete geometric solution to the ESTP; the second, the SMT to Line Heuristic, is a local optimization algorithm used to approximate a solution to our first variation of the ESTP.

2 Background

The Euclidean Steiner Tree Problem originated more than 300 years ago with the curiosity of Pierre de Fermat (1601-1665). “Fermat’s Problem,” as it is now known, asks the following question:

Given three points in a plane, where should we position a fourth point, P , such that the distance from P to all other points is minimal [1]?

Fermat’s original proposition considered only acute triangles in which P would be an interior point, as shown in Figure 2.

Evangelista Torricelli (1608-1647), to whom Fermat posed the problem, solved Fermat’s Problem in 1640. Torricelli constructed a triangle from the three given points and constructed an equilateral triangle off of each resulting edge. He then circumscribed each equilateral triangle. The point P was precisely where each of the three circles intersected, as shown

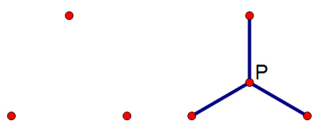


Figure 2: Fermat's Problem.

in Figure 3. Owing to his invention of this method, the point P has since been called the Torricelli point [1].

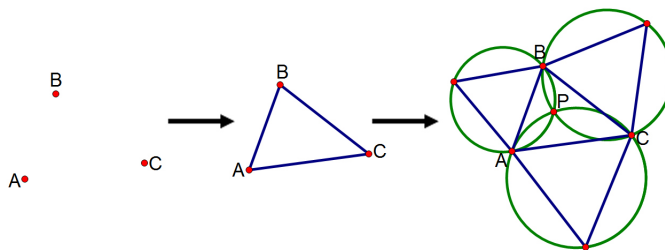
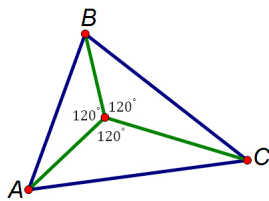


Figure 3: Torricelli's Method performed on the points A, B, and C.

Seven years later, in 1647, Bonaventura Francesco Cavalieri (1598-1647) discovered that all three edges incident to the Torricelli point meet at 120° angles, as shown in Figure 4 [1]. This particular observation has proven extremely useful in locating Torricelli points because

Figure 4: The edges incident to P all intersect at 120° angles.

it allows mathematicians to quickly verify their solutions to Fermat's Problem. Later in this paper, we'll use this characteristic to check solutions to the ESTP in general.

In 1750, Thomas Simpson (1710-1761) discovered a third method for finding the exact location of the Torricelli point. Simpson simply constructed three equilateral triangles from each of the three edges of the triangle formed by points A, B, and C, and connected the vertex of the exterior equilateral triangles to the opposite vertex. These three lines, called *Simpson lines*, intersect precisely at the location of the Torricelli point, P , as shown in Figure 5.

In 1834, Franz Heinen proved that all three Simpson lines are of equal length. Heinen also developed his own method for finding a Torricelli point. Heinen's method, as shown in Figure 6, is as follows:

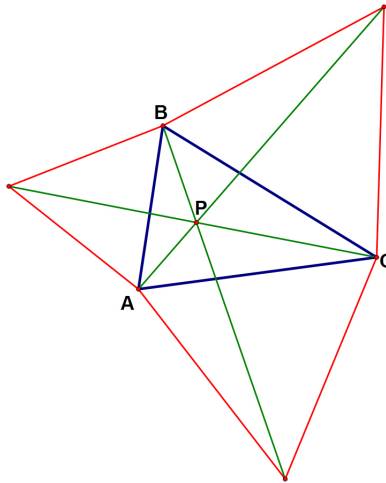


Figure 5: The three Simpson lines intersect at point P , the Torricelli point.

1. First, construct an equilateral triangle from the longest edge of $\triangle ABC$.
2. Second, circumscribe the equilateral triangle.
3. Third, draw a line from the exterior vertex of the equilateral triangle to the opposite vertex as in Simpson's method.
4. Where the circle and the Simpson line intersect is the exact location of the Torricelli point.

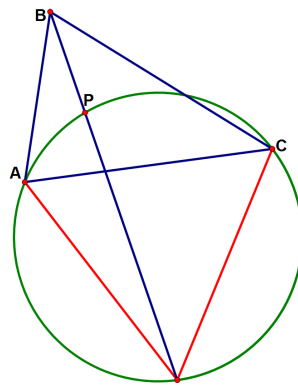


Figure 6: The circle and line drawn intersect at point P , the Torricelli point.

Heinen also proved that the length of any Simpson line is the same length as the sum of the resulting network, $AP+BP+CP$. This result is truly astounding. A single line, so simple in construction, can tell us exactly how long our resulting network will be without actually having to construct the network! Both Heinen's method and the result just mentioned have proven instrumental in our construction of algorithms and research.

In the mid-1880s, Swiss geometer Jakob Steiner (1796-1863) generalized Fermat's problem to include any number of points in the plane. Steiner worked on localized versions of Fermat's problem among a set of points in the hopes of finding a global optimum. For his recognition of the generalization of Fermat's problem, mathematicians Courant and Robbins dubbed the problem "The Steiner Problem" in their 1941 book "What Is Mathematics" [1].

3 Characteristics of Steiner trees

Steiner minimal trees (each an SMT or Steiner tree) are solutions to the ESTP and have certain characteristics that are useful in identifying and searching for SMTs among a set of points, S . Let T be the set of terminal vertices and S be the set of Steiner vertices. The **topology** of a tree is a "connection matrix" in which the connection of each $t \in T$ and $s \in S$ are specified but the locations of all Steiner points are not [3].

A graph has **Steiner topology** if each Steiner point has exactly degree 3, and the each terminal point has degree less than or equal to 3. In 1968, Gilber and Pollak proved that an SMT has Steiner topology. SMTs have distinct properties which we rely upon when searching for solutions to the ESTP. The *angle condition* states that any two edges incident to a Steiner vertex intersect at an angle of 120° or greater. The *degree condition* states that each Steiner vertex has degree 3 and each terminal vertex has degree 3 or less. Together, these conditions imply that all three edges incident to a Steiner vertex intersect at 120° angles [3]. While Cavalieri proved this for Torricelli points in 1647, this condition holds for all Steiner trees with $p \geq 3$ where p is the number of vertices in the graph.

Together, these conditions are particularly useful because they allow mathematicians to immediately check their solutions or, conversely, to limit their search to trees that meet these conditions.

A Steiner topology in which all terminal points have degree one is called **Full Steiner Topology** (FST). In this case, there must be exactly $p - 2$ Steiner points. SMTs must have Full Steiner Topology and accordingly, we restrict our search for Steiner minimal trees to only include minimal spanning trees with FST.

Gilbert and Pollak proved in 1968 that the number of trees with FST grow super-exponentially with respect to the size of the set of terminal vertices. For a set of S Steiner points of order s , the number of full Steiner trees, is $f(s)$.

$$f(s) = 2^{-s}(2s)!/s!$$

Gilbert and Pollak expanded on this equation to find the number of FST as a function of the size of the vertex set, n , and the number of Steiner points, s . For a set of n vertices and s Steiner points, the number of full Steiner trees is

$$F(n, s) = 2^{-s} \binom{n}{s+2} \frac{(n+s-2)!}{s!}$$

While we can restrict our search to trees that have FST, this number grows far too fast to be a truly useful restriction. Table 1 shows just how fast the number of FSTs grows with respect to the size of the vertex set.

n	s	2^n	$\text{fst}(n)$
1	-	2	-
2	0	4	1
3	1	8	1
4	2	16	3
5	3	32	15
7	5	128	945
10	8	1024	2,027,025
13	11	8192	13,749,310,575
15	13	32768	7,905,853,580,625
17	15	131072	6,190,283,353,629,375
20	18	1048576	221,643,095,476,699,771,875
21	19	2097152	8,200,794,532,637,891,559,375
22	20	4194304	319,830,986,772,877,770,815,625

Table 1: The number of FSTs grows super-exponentially with respect to the size of the vertex set.

Because of the complexity of the ESTP, it is believed that it cannot be solved in polynomial time. A problem that can be solved in *polynomial time* if its running time is bounded above by a polynomial expression for a given input size. Algorithms that run in polynomial time, or P , require $p(s)$ operations to complete where p is a polynomial function and s is the size of the input. Most simply, polynomial time algorithms can be run in a reasonable amount of time. If the solution to a given problem can be verified in polynomial time, we say that the problem is in *non-deterministic polynomial time* or NP . A problem that can be solved in polynomial time can also be verified in polynomial time, therefore $P \subset NP$. The ESTP is called *NP-hard*, meaning it is at least as difficult as problems in NP .

4 Basic Steiner Algorithms

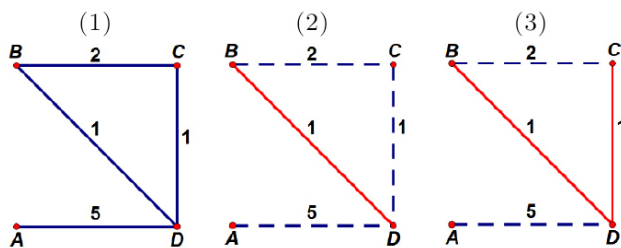
The first step in searching for Steiner trees is to find a minimal spanning tree. Minimal spanning trees often serve as a base for more complicated algorithms but are also a standard

with which we can evaluate the efficacy of algorithms. Both Prim's algorithm and Kruskal's algorithm provide a minimal spanning tree in polynomial time. Prim's algorithm, independently rediscovered by computer scientist Robert C. Prim in 1957, is a greedy algorithm that finds an MST for any connected and weighted graph. For our algorithms, we rely on Prim's algorithm as a base.

PRIM'S ALGORITHM

1. Choose an arbitrary vertex r to be the root of our spanning tree. Set $S = r$. Let the edge set $M = \emptyset$.
2. Choose the edge of least weight that connects any vertex in S to any vertex not in S . Add this edge to the edge set M and the corresponding vertex in \bar{S} to S .
3. Repeat until $\bar{S} = \emptyset$. The tree with vertex set S and edge set M is our MST.

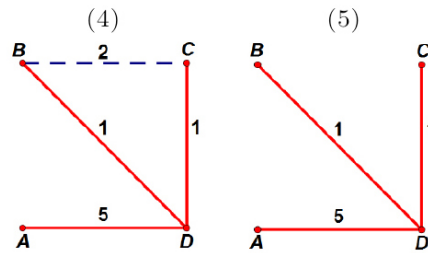
PRIM'S ALGORITHM EXAMPLE



Step 1: Begin with a spanning tree of the vertex set $\{A, B, C, \text{ and } D\}$.

Step 2: Choose an arbitrary vertex. In this case, we have chosen vertex B , so $S = \{B\}$. Choose the edge of least weight that connects B to any other vertex in the set. In this case, \overline{BD} has the least weight, so add \overline{BD} to M and add D to S .

Step 3: Find the next edge of least weight that connects any vertex in S to any vertex not in S . In this case, we are searching for an edge that connects either B or D , the vertices in S , to either C or A , the vertices in \bar{S} . We find that edge \overline{CD} has the least weight. We therefore add \overline{CD} to M and add C to S .



Step 4: We now choose the edge of least weight that connects B , D , or C to A . The only option is \overline{AD} . Thus we add \overline{AD} to M and add A to S .

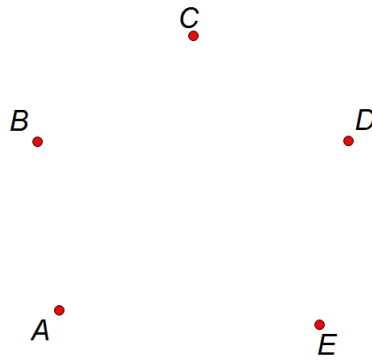
Step 5: At this point, $\overline{S} = \emptyset$ and our edge set M is our MST which has weight 7.

Using an MST, the Steiner insertion heuristic algorithm returns an SMT approximation that meets the angle condition. Developed by Dreyer and Overton, the Steiner insertion heuristic follows [6].

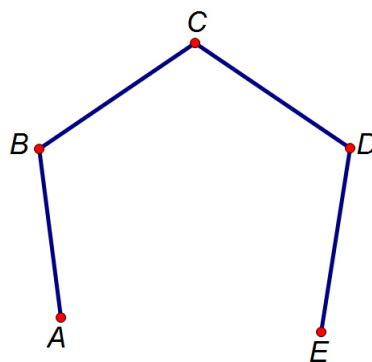
STEINER INSERTION HEURISTIC

1. Find the minimal spanning tree.
2. For each edge (t_x, t_y) connecting fixed terminal vertices t_x and t_y do the following:
 - (a) Find the edge (t_y, t_z) that meets (t_x, t_y) at the smallest angle, where t_z can be either a fixed point or a Steiner point.
 - (b) If this angle is less than 120 degrees then:
 - i. Place a new Steiner point s_n on top of t_y where n represents the number Steiner vertex that is being inserted.
 - ii. Remove the edges (t_x, t_y) and (t_y, t_z) . These edges will no longer be considered for the loop of Step 2.
 - iii. Add the edges (t_x, s_n) , (t_y, s_n) , and (t_z, s_n) .
3. Run the local optimization algorithm, as developed by Dreyer and Overton, on the tree with its new topology.

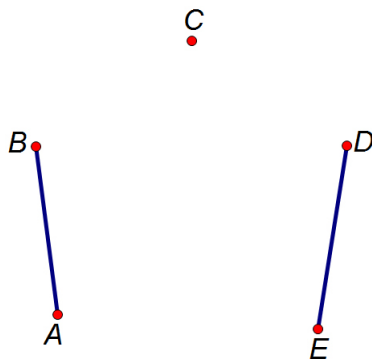
STEINER INSERTION HEURISTIC EXAMPLE



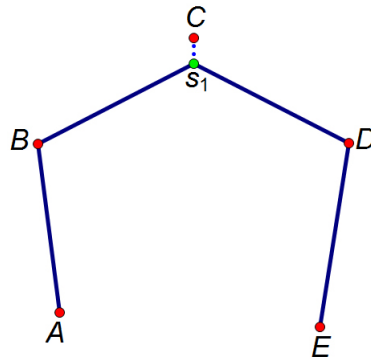
Step 0: The set of terminal vertices A, B, C, D, E .



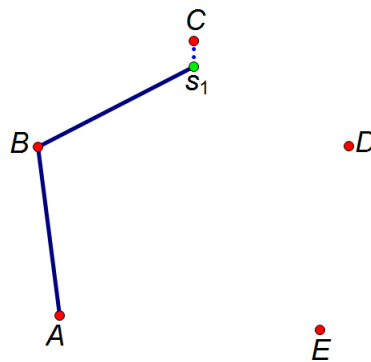
Step 1: Form a MST among the vertex set.



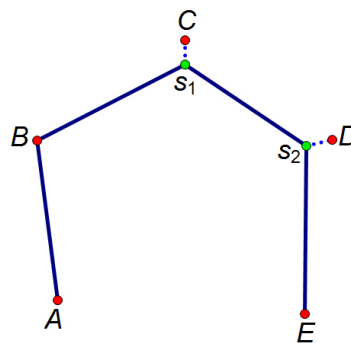
Step 2: Edges \overline{BC} and \overline{CD} meet at the smallest angle, so we will begin with these edges. Since this angle is less than 120 degrees, we then remove these two edges.



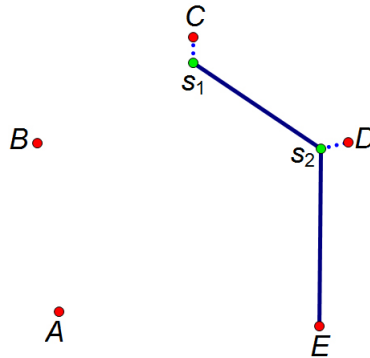
Step 3: Insert Steiner vertex s_1 on top of vertex C and connect s_1 to vertices B and D. For the sake of the reader, s_1 is moved slightly to show vertex C. Correspondingly, the blue dotted line has weight 0. For the duration of the algorithm, we'll use this representation to make it easier to follow.



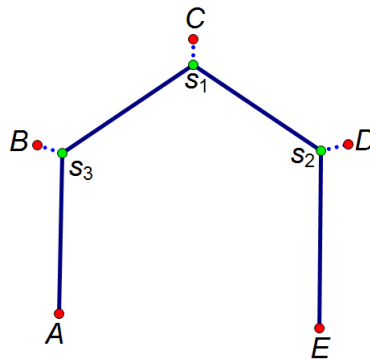
Step 4: We now find the next set of edges that meet at the smallest angle but do not consider the edges that have already been removed \overline{BC} and \overline{CD} . In this case, the next set of edges is \overline{CD} and \overline{DE} . We then delete edges \overline{CD} and \overline{DE} .



Step 5: Insert Steiner vertex s_2 and add edges $\overline{s_1E}$ and $\overline{s_1s_2}$.



Step 6: We now find the next set of edges that meet at the smallest angle. In this case, the next set of edges is \overline{AB} and \overline{BC} so delete these edges.



Step 7: Insert Steiner vertex s_3 and add edges $\overline{s_3A}$ and $\overline{s_3s_1}$. Since all edges now meet at 120 degrees, our example is finished and we have our final SMT approximation.

As noted, MSTs can provide a lower bound with which we can evaluate our algorithms for finding SMTs. We know for sure that the length of any given SMT has to be less than the MST for the same set of terminal vertices. Let $l(N)$ be the length of any given network. Given a set of points terminal vertices, T , let $MST(T)$ be the MST among T and $SMT(T)$ be the SMT among T . If we compute the ratio $l(SMT(T))/l(MST(T))$ for all sets of terminal vertices, we find that the minimum value, $p = \frac{\sqrt{3}}{2} \approx 0.866$, as proven by Du and Hwang [5]. For $p = 3, 4,$ and 5 , the Steiner ratio is in fact $\frac{\sqrt{3}}{2}$, as shown by Du [4].

5 Discussion of Fixed Steiner Line and Points

Now, we'll consider our first variation on the classic ESTP. How should we create a minimal spanning network if we are given a set of points and a line that cannot be moved? We can easily solve this problem for any case of $p = 2, 3, 4, 5$ where p is the number of points on one

side of the line. If we wanted to simply connect two points to a line, we could create a Steiner network among the two vertices and the line. Depending on the placement of the points, however, it may be more efficient to connect the points to the line via two perpendicular line segments. If this is the case, then creating a Steiner network is not always optimal. This is shown in Figure 7.

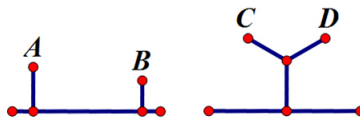


Figure 7: Terminal vertices A and B are most efficiently connected to the line using perpendicular line segments while vertices C and D use a SMT.

It is not immediately obvious, however, that such a result holds for $p > 2$. In Figure 8, we've shown three different optimal results given three separate and different orientations of three points. There are other orientations, of course, but these represent the insertion of 1, 2, and 0 Steiner points, respectively, to create an optimal network.

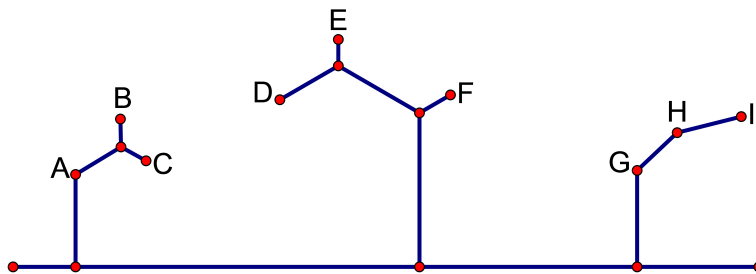


Figure 8: Only the center graph has FST. Note that the original vertices for each graph are labeled and that the Steiner points are unlabeled.

So how might we tell exactly when it is optimal to drop perpendicular line segments and when it is optimal to create a Steiner network? It turns out that we can use some of Heinen's results to figure this out. We have found a clean geometric method for finding out exactly when we should form a Steiner network to connect two points to a line and when we should forgo the Steiner network and simply drop perpendicular segments to the fixed line. To demonstrate this result, we'll begin by reviewing some variations of Heinen's results for small vertex sets. We can assume without loss of generality that all points are on the same side of the line.

Proposition 1. *For any two arbitrary points A and B and a line in a plane, let X be the third vertex of an equilateral triangle with base AB that points away from the line. Let C be the circle that circumscribes the equilateral triangle, $\triangle ABX$. The optimal placement of a Steiner point S is at the intersection of a perpendicular line connecting X and the fixed line and circle C .*

Recall Heinen’s method for finding the Torricelli point discussed in the Background section. This result follows from the same logic.

Proof. Consider the points A and B and the line shown in Figure 9.

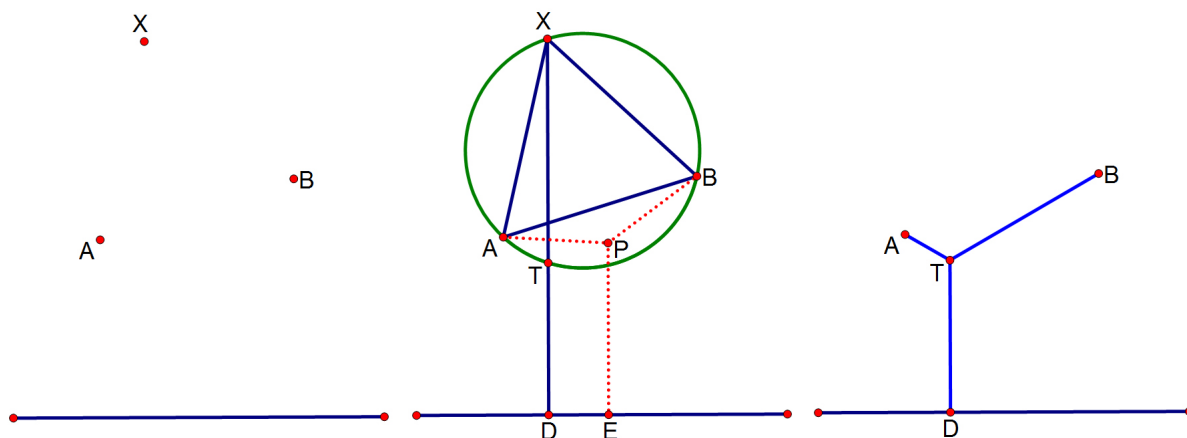


Figure 9: Heinen’s Method for Finding an SMT with two vertices and a line, l .

We will demonstrate a method for finding the Steiner point to connect A and B to the line, l . First, we’ll construct equilateral triangle $\triangle ABX$ off of segment AB and circumscribe a circle about $\triangle ABX$. Let D be any point of intersection of line l and a line segment that connects X to l .

Consider any point P in the plane. For point P to be a Steiner point, it must have full Steiner topology. So point P must have degree 3 and all adjacent edges must form 120° angles. Point P must then be connected to l by a line segment perpendicular to l which intersects l at some point, E . For $\angle APB$ to be 120° , point P must lie on the circle circumscribed around $\triangle ABX$. By Ptolomey’s Theorem, we know that

$$(PB)(AX) + (PA)(XB) \geq (XP)(AB). \tag{1}$$

Note that $\triangle ABX$ is equilateral, so $\overline{AX} \cong \overline{BX} \cong \overline{AB}$.

We can then simplify equation (1) to obtain

$$PX \leq PB + PA. \tag{2}$$

By the triangle inequality, we also know that

$$XD \leq XE \leq XP + PE. \tag{3}$$

If we add equations (2) and (3) together and subtract the PX from both sides, we are left with

$$XD \leq PB + PE + PA. \tag{4}$$

So given any placement of point P and E , $XD \leq PB + PE + PA$, with equality only if P is on the circle and XE is perpendicular to line l . This means that P must be situated at point T , where T is the intersection of a perpendicular line which connects X and l , and circle C , to obtain the optimal configuration. The resulting minimal Steiner Network consists of the set of points $\{A, B, T, D\}$.

□

In general, this method will work for any set of 2 points and a line. It is possible, however, that we may arrive at an external, and thus extraneous, solution. Using this method and the fact that the length of each Simpson line is equal to the length of the resulting SMT, we can easily find when constructing a SMT will be more advantageous than creating a pair of perpendicular line segments.

Theorem 2. *For any two points, A and B , and a line, l , if the third point of an equilateral triangle pointing toward line l that is formed from the two points A and B is below the Steiner line, the optimal minimal network connects the two points A and B to line l via perpendicular line segments.*

If the third point of an equilateral triangle is above the fixed line l , the optimal resulting network will have a single Steiner point and will have full Steiner topology.

If the third point of an equilateral triangle lies on the fixed line l , both networks are of equal weight.

Proof. Consider any two points A and B and a fixed line l such that the third vertex, C , of an equilateral triangle constructed off of segment \overline{AB} lies on line l . We will construct an SMT that connects A , B , and l via Steiner point S , as described in Proposition 1. Call this SMT Network 1. Denote by X the new vertex in the equilateral triangle constructed off of \overline{AB} pointing away from line l . Call the point where the perpendicular line that connects X to line l intersects point D . From Proposition 1, we know that the length of SMT Network 1 is equivalent to the length of \overline{XD} .

The only other possibility for a minimal spanning tree that would connect A and B to line l would be that which connects A and B to line l via independent line segments that are perpendicular to line l . Call the point of intersection where the perpendicular line segment connects A to l point A_l and the point of intersection where the perpendicular line segment connects B to l point B_l . Thus we have perpendicular line segments $\overline{AA_l}$ and $\overline{BB_l}$. Call this minimal spanning tree Network 2. For our consideration, the length of Network 2 shall be the sum of the lengths of the perpendicular line segments $\overline{AA_l}$ and $\overline{BB_l}$.

We will focus on the specific case where the third point, C , lies on line l . The other cases can be considered as shifts from this case. We will show that as C moves up, Network 1 is of less weight than Network 2, and as C shifts down, Network 2 becomes preferable.

In Figure 10, we've constructed points A and B such that the third point of the equilateral triangle pointing toward $\overline{A_l B_l}$ lays on $\overline{A_l B_l}$. This third point is represented as point C . The relative position of points A and B are of no consequence provided that the third point C of the equilateral triangle lays on $\overline{A_l B_l}$. We've also constructed $\overline{A'B'}$ such that it is parallel to $\overline{A_l B_l}$ and passes through point X . By constructing perpendicular line segments from

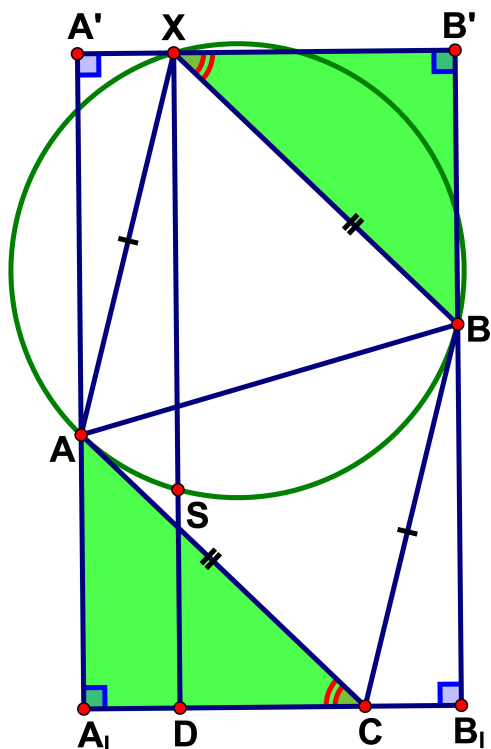


Figure 10: When C is on line l , Network 1 and Network 2 are of equal weight.

A and B to line $\overline{A'B'}$, we can construct similar triangles $\triangle AA_1C$ and $\triangle BB'X$. Since the triangles are similar, $\overline{XB} \cong \overline{AC}$ and $\overline{AA_1} \cong \overline{BB'}$. The length of the perpendicular network, $\overline{AA_1} + \overline{BB_1}$ is equal to the length of $\overline{B'B_1}$. We already know that \overline{XD} is equal to the length of the resulting Steiner network. We can now compare the length of $\overline{B'B_1}$ and the length of \overline{XD} to determine and compare the weights of each individual solution. Since both $\overline{B'B_1}$ and \overline{XD} are perpendicular to $\overline{A'B'}$ and $\overline{A_1B_1}$, $\overline{B'B_1} \cong \overline{XD}$. Thus when the third point of an equilateral triangle pointing toward the line lies on that fixed line, the perpendicular network and Steiner network are of equal weight.

So if C sits on the line, the weight of Network 1 is equal to that of Network 2. If the points A and B are shifted upward, X and C will also be shifted upward. The length of the Steiner minimal network, Network 1 will be increased by only the amount shifted, while the length of the Network 2, the perpendicular network, will be increased by twice as much. The converse is also true.

Thus if the third point, C , sits above line l , the optimal network will have a single Steiner point and will have full Steiner topology. If C sits below line l , the optimal network is formed by connecting points A and B to line l via perpendicular line segments. If C sits on line l , both networks are of equal weight.

□

This result is perhaps the most important in our research. This means that the Steiner

network formed to connect two points to a line is not necessarily the network of least weight. It should be noted, however, that we are not counting the weight of the Steiner line in our resulting network.

6 Discussion of Placement of Optimal Steiner line

In this discussion we seek to answer our second research question. Imagine that we are trying to place a highway so as to minimize the remaining amount of road to be constructed that connects cities to that highway. We want to figure out the optimal placement of that highway. For $p \leq 3$, we have found exactly where this placement should be. We'll demonstrate the case in which $p = 3$, excluding the null cases of $p = 1$ and $p = 2$. (Note that if $p \leq 2$, we may place the line such that it passes through each vertex in the vertex set. Such a result is already optimized since the line spans the entire network.)

Proposition 3. *For any three arbitrary points in a plane, the position of a Steiner line giving a minimal optimal network is that which passes through the two points that are furthest apart.*

Proof. We will show that in general, for three arbitrary points in a plane, the best way to position a straight line is to connect the two points that are furthest apart. Consider the three points in a plane, A , B , and C shown in the figure below. There are five possible arrangements of points A , B , and C and a line to consider.

Case 1: If points A , B , and C line on the same line, the result is obvious.

Case 2: The line l is outside $\triangle ABC$. Clearly all distances from the points to the line decrease as we move it towards points A , B , and C . We can easily conclude that this placement of the line cannot be optimal. Observe Figure 11.

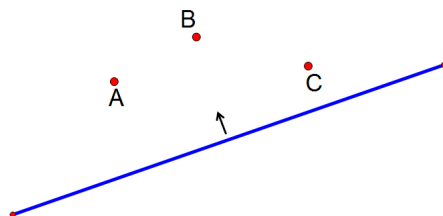


Figure 11: Case 2: The distance from the points to the line decreases if the line is shifted toward the points.

Case 3: Consider any line that intersects any two sides of $\triangle ABC$ but does not intersect any of the points. In other words, the line passes between the points. We can clearly decrease the total distance of the connecting lines by translating our new line toward toward two points

and away from the third. Observe Figure 12.

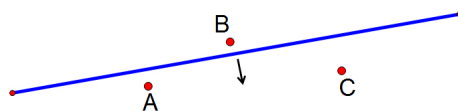


Figure 12: Case 3: The distance from the points to the line decreases if the line is shifted or pivoted toward any two points.

Case 4: Consider any line that passes through a single vertex, A , but otherwise lays outside of the $\triangle ABC$. By pivoting the line on vertex A , we can move it closer to both B and C . Eventually, the line will intersect at least one other point, B , and the remaining distance will be only between the line and point C . Observe Figure 13.

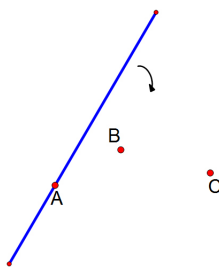


Figure 13: Case 4: Consider what what would happen if we were to pivot the line.

Case 5: Similarly, if a line intersects a single vertex A and intersects the edge formed by connecting B and C , we can form a smaller network by pivoting the line to whichever vertex is farthest from A . In fact, we can prove this result in general. We'll show that the optimal placement is for the line to connect the two points furthest apart.

Let $d(x, y)$ be the distance between any two objects. Construct points B and C such that $d(A, B) < d(A, C)$. Next, construct an arbitrary line \overline{VX} which intersects vertex A and edge \overline{BC} (not shown). We'll construct a line segment H_1 between B and \overline{VX} such that the length of H_1 is equal to $d(B, \overline{VX}) = h_1$. We'll also construct line segment H_2 between C and \overline{VX} such that the length of H_2 is equal to $d(C, \overline{VX}) = h_2$. Next, we'll construct segment \overline{WY} parallel to \overline{VX} such that $d(\overline{VX}, \overline{WY}) = h_2$. Because $d(\overline{VX}, \overline{WY}) = h_2$, we can construct a line segment H'_2 adjacent to segment H_1 , such that $H_2 \cong H'_2$, H'_2 is parallel to H_2 , and H'_2 intersects \overline{WY} at point F , and $h_2 = h'_2$. We'll consider line \overline{UZ} which connects vertices A and C . The shortest distance from B to line \overline{UZ} is the perpendicular segment with length d_1 , which intersects \overline{UZ} at point D . By construction, \overline{BD} intersects \overline{UZ} at a 90° angle. By construction, $\angle BDF > \angle BDC$, so $\angle BDF > 90^\circ$. This means that the resulting triangle formed by \overline{BD} , \overline{BF} , and \overline{DF} is obtuse. The side of length $h_1 + h_2$, which is formed by H_1 and $\overline{H'_2}$ is opposite the obtuse angle, $\angle BDF$. Therefore it must be the case that $d_1 < h_1 + h_2$.

Thus the length of the overall network is minimized by placing the Steiner line such that it passes through points A and C . We can clearly see this is Figure 14.

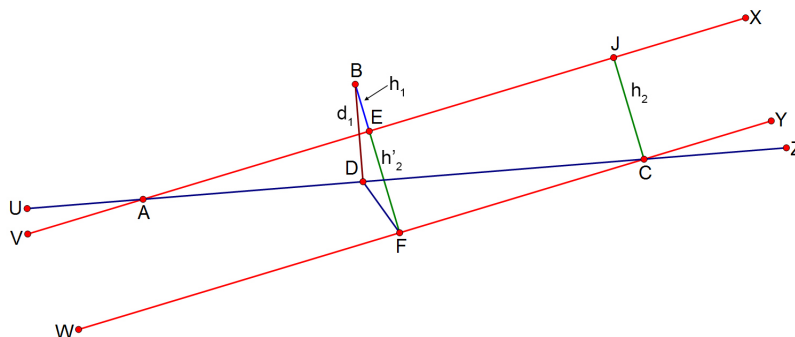


Figure 14: The optimal placement is that which connects the two points of furthest distance.

So any other placement of a line will result in a greater distance of remaining road. Therefore, the optimal placement of a line connecting three points in a plane is that which connects the two points of furthest distance. \square

We expect that by employing the same logic utilized in Proposition 3, for any set of points in a plane, a Steiner line must pass through at least two of the given points. However, the proof of this is beyond the scope of this paper.

7 Searching for Optimal Solutions

In our search for optimal solutions, we run into a few problems. Since the ESTP is NP -hard, we are faced with a great challenge when merely trying to construct Steiner trees on one side of a line. Further, we have to identify candidate sets of points that should be in the SMT and those which should be excluded in favor of an optimal solution. The SMT searcher should not yet be deterred, however, as we have arrived at two useful algorithms which can get us closer to an optimal solution for relatively small sets of vertices. The first of our algorithms takes any basic SMT approximation that has Steiner topology and returns an exact SMT with FST. We believe that this algorithm, which we've dubbed The Steiner Correction Algorithm, was likely discovered by Heinen. We have found no record of this, but it is powered by all of Heinen's results. Recall that we can quickly and easily find an SMT approximation with Steiner topology via the Steiner Insertion Heuristic described above. However, we note that Dreyer and Overton's Incremental Optimization Heuristic [6] is a marked improvement on the SI Heuristic, as the SI Heuristic can fail to insert critical Steiner points. For large random sets of points, however, the SI Heuristic is nearly as good. We include the Incremental Optimization Heuristic in the following algorithm.

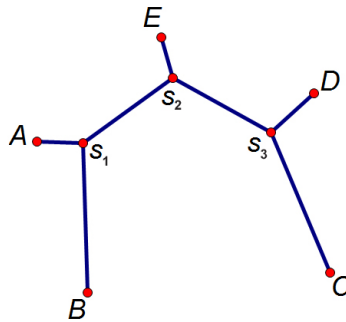
STEINER CORRECTION ALGORITHM

Let T be a set of terminal points and $S = \{s_1, s_2, \dots, s_n\}$ be a set of approximate Steiner points from the Incremental Optimization Heuristic.

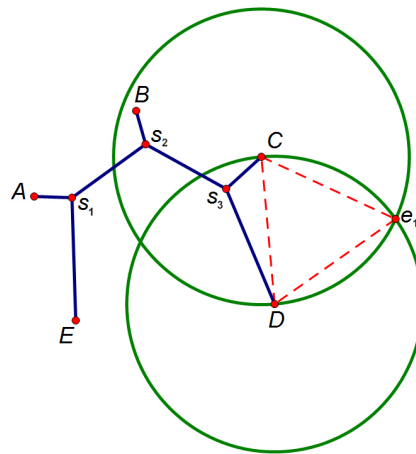
1. Find any two terminal points $t_x \in T$ and $t_y \in T$ such that t_x and t_y are any two terminal points connected via a Steiner vertex, $s_i \in S$, where i represents the number Steiner vertex being considered. Connect these two terminal vertices and form an equilateral triangle off this new edge. Call the new point of the equilateral triangle $e_i \in E$, where E is the set of auxiliary vertices of equilateral triangles and i represents the number vertex from set E being considered.
2. Delete the two connected terminal vertices and the associated Steiner vertex. Connect the new point, e_i to the Steiner vertex that was adjacent to s_i .
3. Repeat Steps 1 and 2 until there is only one terminal vertex remaining.
4. Repeat Steps 1 and 2 but include any $e \in E$ and $t \in T$ connected via a Steiner vertex.
5. Connect the two remaining vertices in E formed from the equilateral triangles. Call this line l .
6. Reinsert the deleted vertices and edges in the order that they were deleted. Circumscribe the equilateral triangle formed by the edge connecting the reinserted vertices. Where l intersects the circle is the exact location of the associated Steiner point.
7. Connect the new Steiner point to the remaining $e_i \in E$. Call this new line l .
8. Repeat Steps 5 and 6 until the exact location of all Steiner vertices has been found.
9. Form final SMT.

While complex in words, this algorithm can be easily understood with the accompanying diagrams and explanation. This algorithm begins to provide useful results by Step 5. Note that the length of line l found in Step 5 is equivalent to the length of the resulting Steiner network. This means that before we are halfway done with the algorithm, we know one of the most important pieces of our final result! Below we present an example of our algorithm.

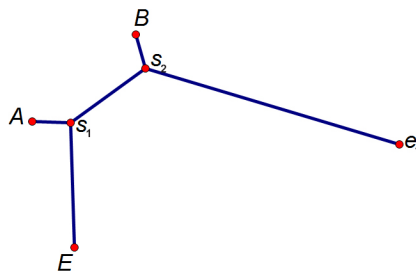
STEINER CORRECTION ALGORITHM EXAMPLE



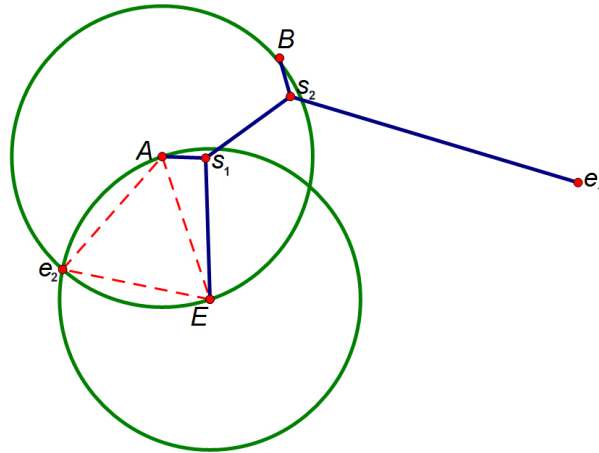
Step 1: The terminal points $T = ABCDE$ are connected via approximations of Steiner vertices from the Steiner Insertion Heuristic Algorithm.



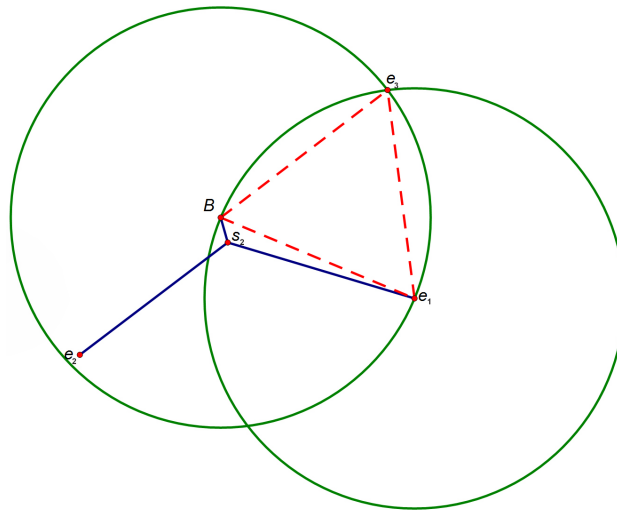
Step 2: Find and connect any two terminal vertices connected by a Steiner vertex. Call the point of the equilateral triangle formed by this edge, e_1 .



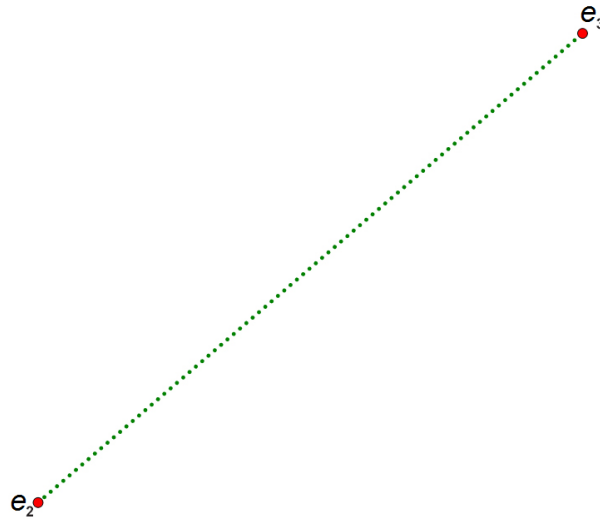
Step 3: Delete the two connected terminal vertices, C and D , and the associated Steiner vertex, s_3 . Connect the new point, e_1 , to the Steiner vertex adjacent to the one just deleted, which in this case is s_2 .



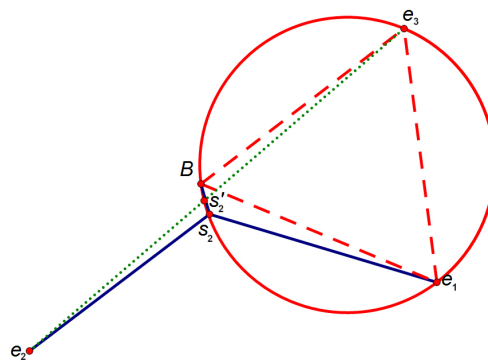
Step 4: Repeat Steps 2 and 3 until there is only one terminal point remaining. Here, we add e_2 .



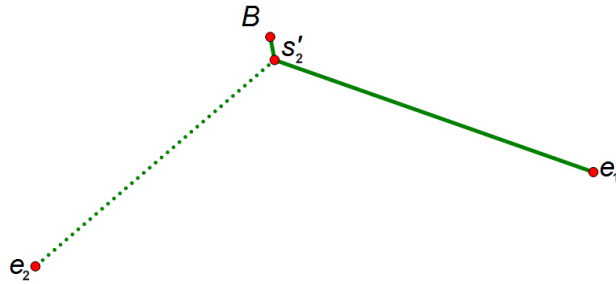
Step 5: Repeat Steps 2 and 3, but include any $e \in E$ and $t \in T$ connected via a Steiner vertex. Here, we delete terminal vertices A and E , and the associated Steiner vertex, s_1 . We then connect e_2 to the Steiner vertex adjacent to s_1 , which in this case is s_2 .



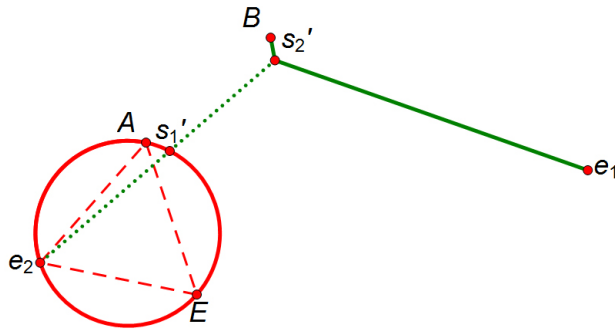
Step 6: After removing vertices B , e_1 , and s_2 . Connect the two remaining e vertices formed from the equilateral triangles. Call this line l . Note that the length of line l equivalent to the length of the overall resulting Steiner minimal network.



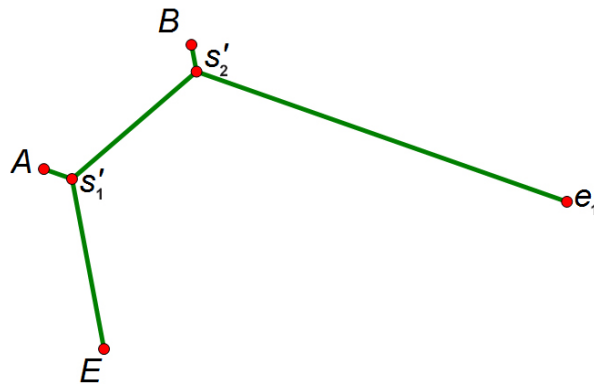
Step 7: Reinsert deleted vertices and edges in the order that they were deleted. Here, we reinsert vertices B , e_1 , and s_2 . Circumscribe the equilateral triangle formed by the edge connecting them. Where line l intersects the circle is the exact location of the associated Steiner point.



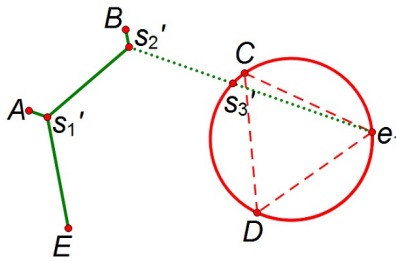
Step 8: Connect the new Steiner point to the remaining $e \in E$. Call this new line l .



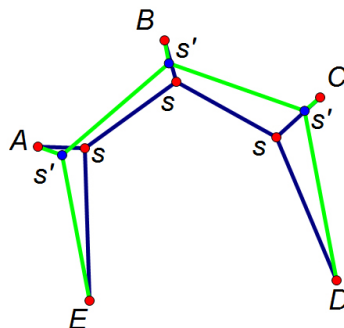
Step 9: Repeat Steps 6 and 7 until the exact location of all Steiner vertices have been found. In this graphic, we are repeating Step 6 and reinserting vertices A and E and the associated Steiner vertex, s_1 .



Step 9(i): Repeating Step 7.



Step 9(ii): Repeating Step 6. Here, we reinsert vertices C and D .



Step 10: The approximate and exact Steiner trees are overlaid. The approximate SMT has blue edges and the exact SMT is green with Steiner points s' .

Our second algorithm is a heuristic algorithm that takes a set of vertices and connects them to a given Steiner line. The algorithm seeks local solutions in the hope of finding a global optimum. We rely on the Steiner Correction Algorithm to correct local approximations of SMTs.

SMT TO LINE HEURISTIC

Input: A set of terminal points, T , and a line l . $|T| = p$.

Output: A Steiner minimal tree and auxiliary perpendicular segments that span T and connect all points in T to l .

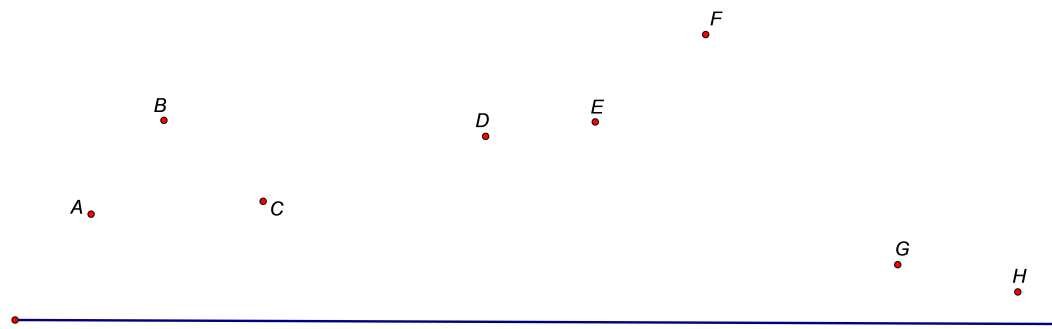
We can assume without loss of generality that all points are on the same side of the line, since we'll run this heuristic twice—once for each set of points on either side of the line.

1. Rotate the terminal vertices and the line so that the line is horizontal and form a MST on the set of terminal vertices. Drop a perpendicular line segment from the terminal vertex closest to l . Set this as *currenttree*.
2. Order all vertices in T by their x -coordinate from smallest to largest.

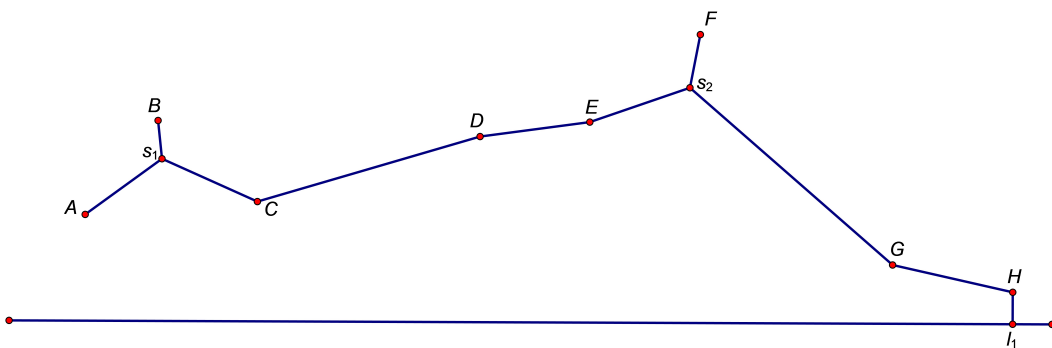
3. Let $\pi_1, \pi_2, \dots, \pi_k$ be pairs of vertices where π_i is a pair of vertices such that they are either 1) connected or 2) connected via a Steiner vertex. For each π_i , evaluate the local optimum using Theorem (2).
 - (a) If the resulting tree is of less weight than *currenttree*, set the new tree to *currenttree*. If the resulting tree is not of less weight, move on to π_{i+1} .
4. Our final *currenttree* is our final network.

As was the case with the Steiner Insertion Heuristic, this algorithm is far easier to understand with the accompanying diagrams.

SMT TO LINE HEURISTIC EXAMPLE 1, PART 1

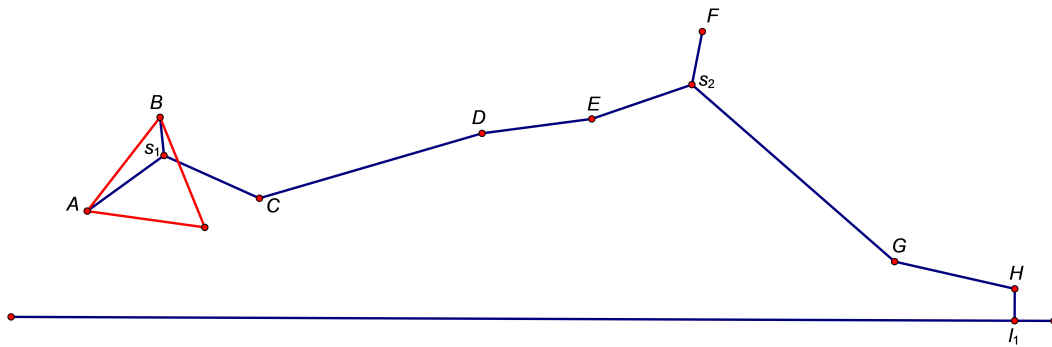


Step 0: $T = \{A, B, C, D, E, F, G, H\}$.

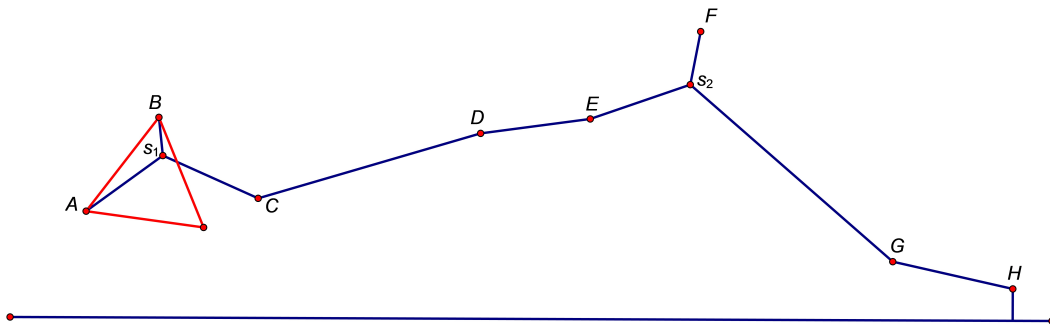


Step 1: Form an SMT among the vertices of T . Here, we have dropped a perpendicular line segment from H to l , intersecting at point l_1 . This SMT has length 38.44.

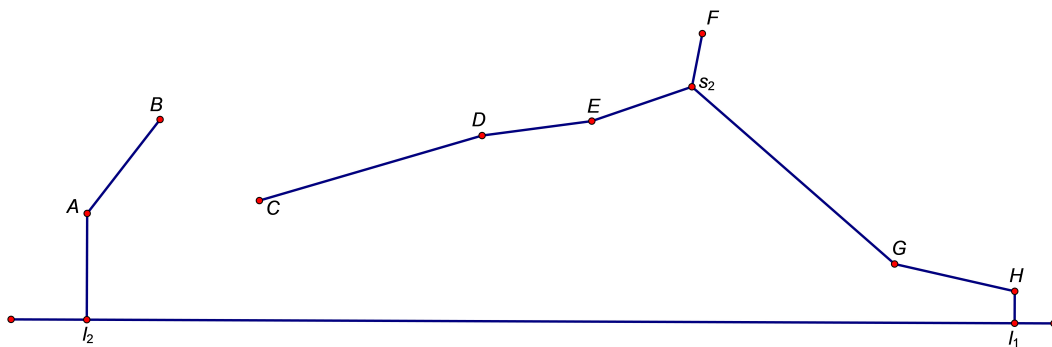
Note: Step 2 is not necessary in this case.



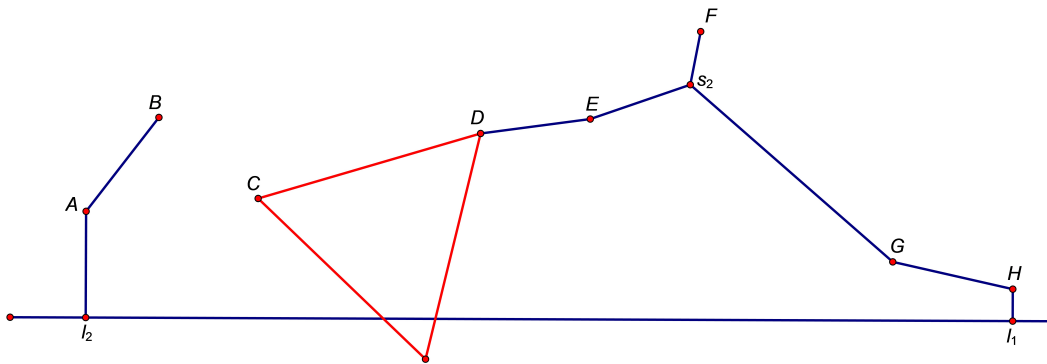
Step 3: Evaluating $\pi_1 = \{A, B\}$ using Theorem (2) for a local optimum.



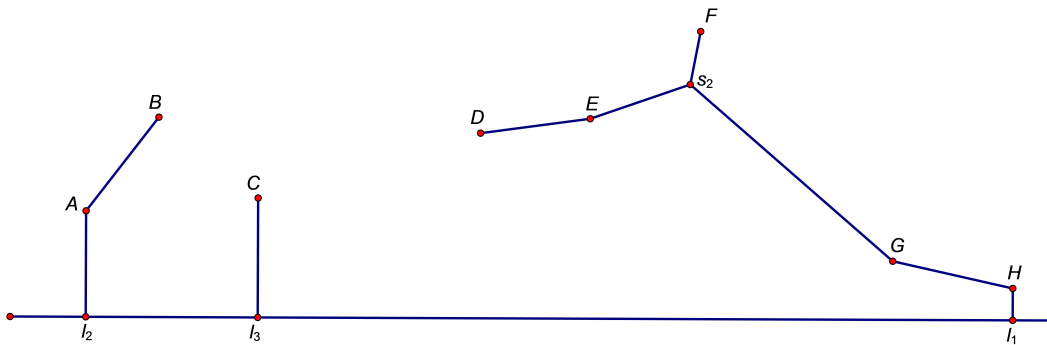
Step 3a: Evaluating $\pi_2 = \{C, B\}$ using Theorem (2) for a local optimum.



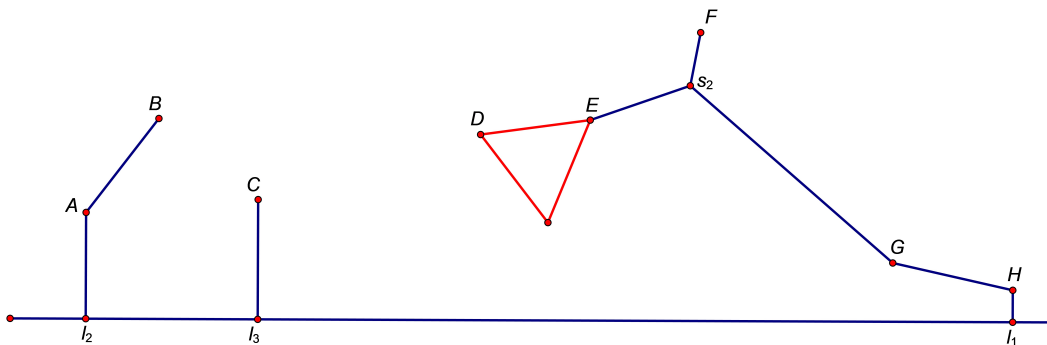
Step 3b: Locally correcting our SMT according to the results of Theorem (2) in the previous step. The network currently has length 38.03



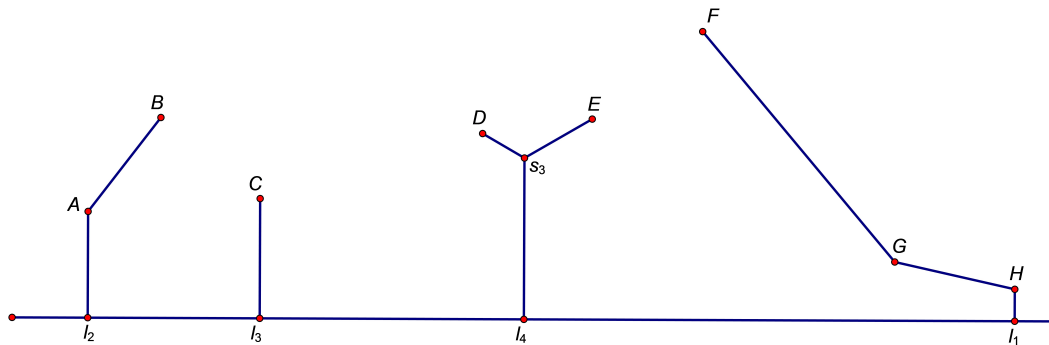
Step 3c: Evaluating C and D using Theorem (2) for a local optimum.



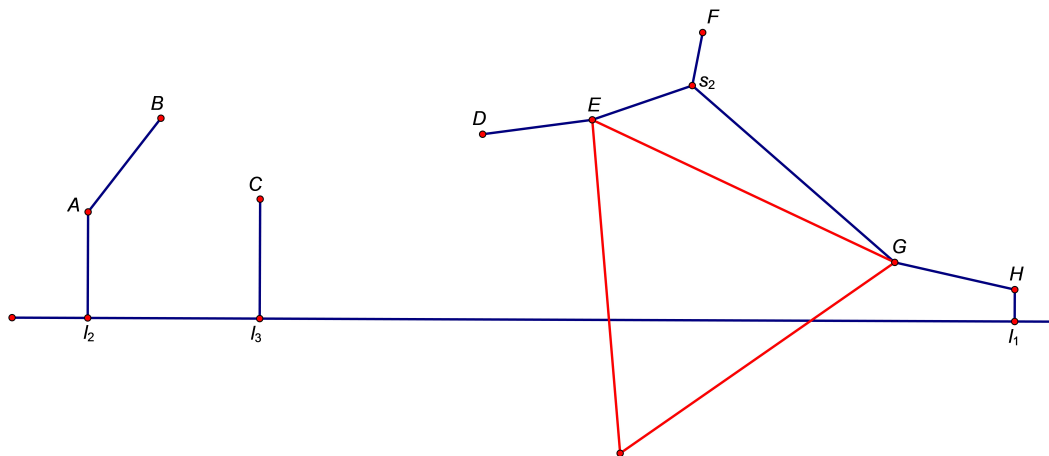
Step 3d: Locally correcting C and D . At this point, *currenttree* has weight 34.32.



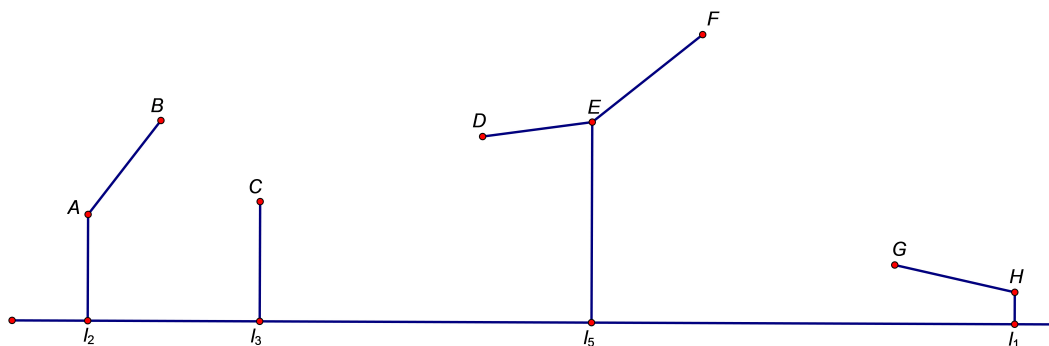
Step 3e: Evaluating D and E using Theorem (2) for a local optimum.



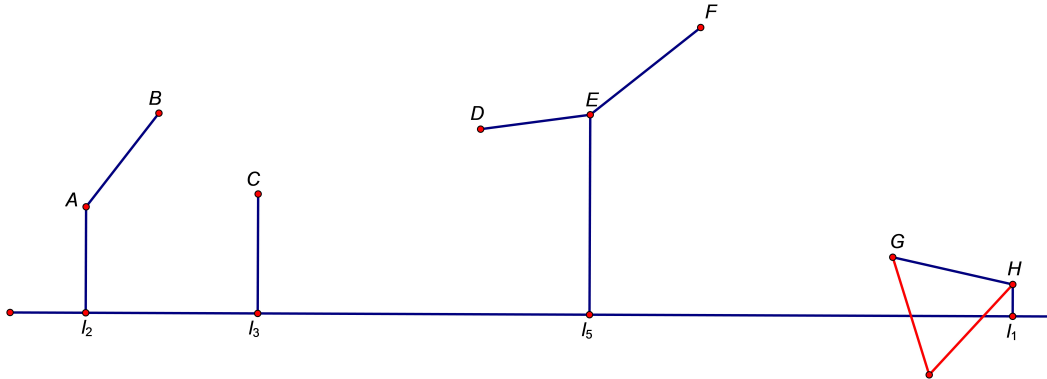
Step 3f: Our local correction from Step (3e) returns a tree which has weight 35.89. Since this is greater than the weight of *currenttree*, we reject this correction and continue the algorithm on vertices *E* and *F* without reassigning *currenttree*.



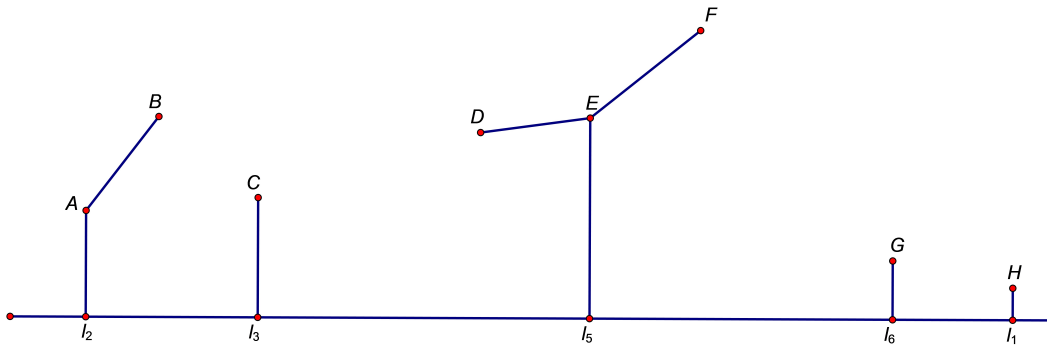
Step 3g: An evaluation of *E* and *F* naturally concludes that they should be connected via a Steiner vertex so we have no change. An evaluation *E* and *G* finds that they should connect to *l* via perpendicular lines.



Step 3h: At this point, *currenttree* has length 31.41. Note that the network formed by D, E, F , and l_5 do not form an SMT.



Step 3i: Evaluating G and H using Theorem (2) for a local optimum.

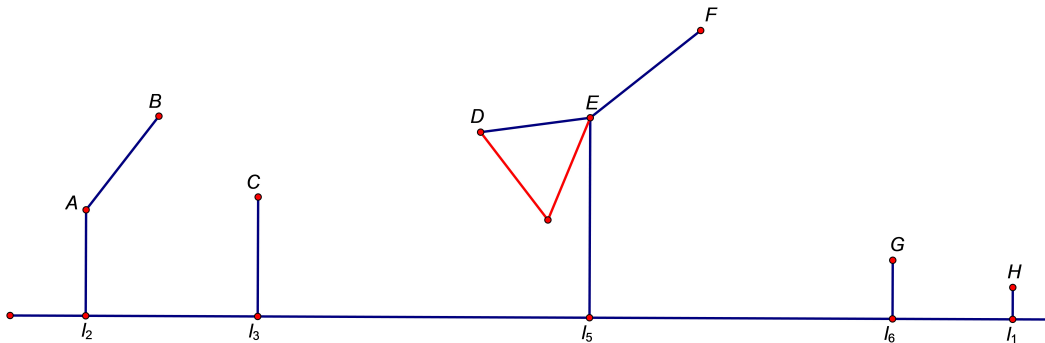


Step 4: After locally optimizing G and H according to Theorem (2) we have constructed a MST that connects the network to l . In this case, we have an MST of weight 29.28. Our final *currenttree* is 76.2% the weight of the original MST.

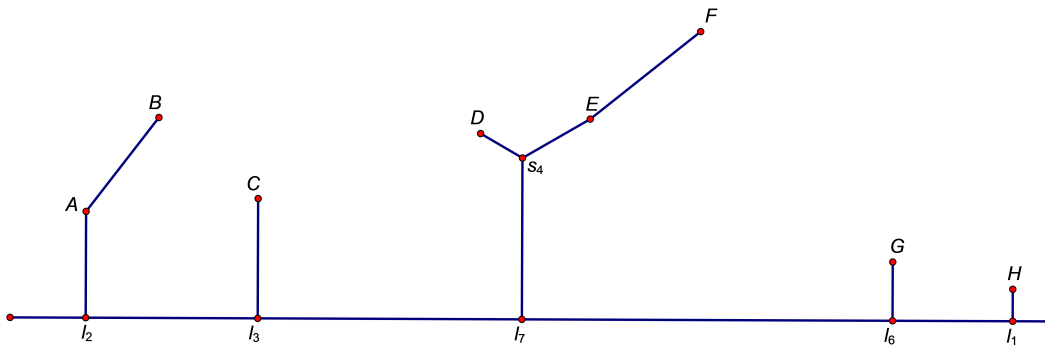
When run iteratively by using the final *currenttree* from the previous iteration as a starting point for the next loop of the algorithm, it will correct many of its own errors.

In SMT to Line Heuristic Example, Part 2, below, we have demonstrated how the second iteration of this algorithm will produce an MST with FST.

SMT TO LINE HEURISTIC EXAMPLE 1, PART 2



Part 2, Step 3: Begin with final *currenttree* from Step 4 in Part 1 of the example. We will now evaluate A and B according to Theorem (2). We know the result of this evaluation from Part 1 of the example. The next pair of vertices that is connected or are connected via a Steiner vertex are vertices D and E . We know the result of this evaluation from Part 1 of the example, so we will construct a local correction using Theorem (2).



Part 2, Step 4: Our final *currenttree* of Part 2 has FST and has weight 28.51. Our final *currenttree* in Part 2 is 74.2% the weight of the original MST from Part 1. (Note that s_4, E , and F do not form an SMT because $\angle s_4EF > 120^\circ$.)

SMT TO LINE HEURISTIC EXAMPLE 2

For this example, we demonstrate the algorithm on a famous configuration of points in Figure 15 and Figure 16. Illustrations for Steps 2 and 3 are not included.

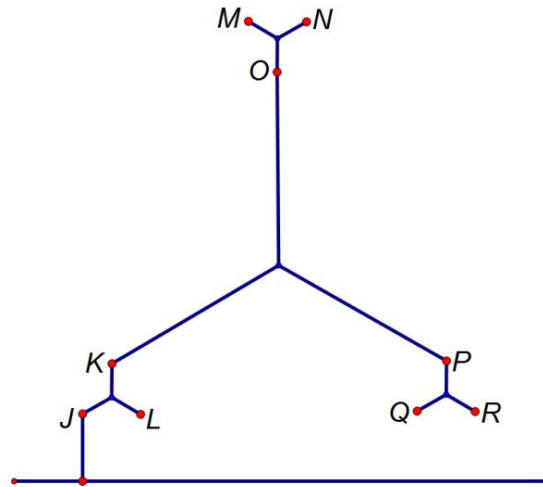


Figure 15: Step 1: We construct an SMT among $T = \{J, K, L, M, N, O, P, Q, R\}$ and connect the SMT to the line via the shortest perpendicular line segment. Note that the Steiner vertices are colored blue. This *currenttree* has a total weight of 1209.

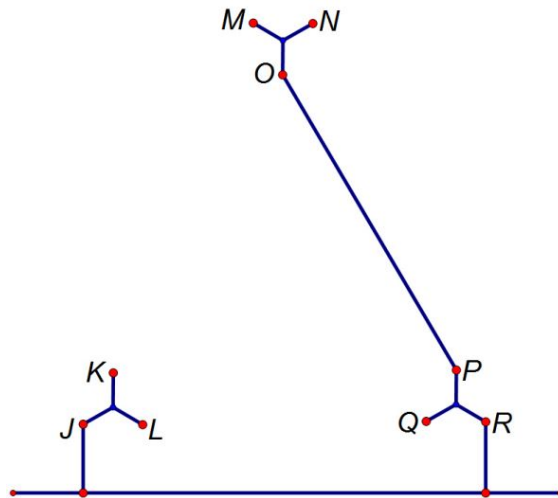


Figure 16: Our final *currenttree* has weight 985, a 19.6% decrease in weight from our original SMT.

While it is a heuristic algorithm, when run iteratively by using the final *currenttree* from the previous iteration as a starting point for the next loop of the algorithm, it will correct many of its own errors. In the Example 1 and Example 2, we were able to decrease the

lengths of the Steiner networks by 25.83% and 19.6%, respectively. Bear in mind that the Steiner network by itself has already shrunk the minimal spanning tree by a maximum of 13.4%.

For those tasked with creating a physical network, be it pipelines, networks, or roads, these findings can result in enormous savings. We hope that future researchers will use the various theorems, propositions, and algorithms presented in this paper to contribute to the field of graph theory.

References

- [1] D. Cieslik, *Steiner Minimal Trees*, Kluwer Academic Publishers, Boston, 1998.
- [2] M.R. Garey, R.L. Graham, and D.S. Johnson, The complexity of computing Steiner minimal trees, *SIAM Journal of Applied Mathematics* **32** (1977), 857-859.
- [3] E. N. Gilbert, H. O. Pollak, Steiner Minimal Trees, *SIAM Journal of Applied Mathematics* **16** (1968).
- [4] D. Z. Du, The steiner ratio conjecture is true for five points, *Journal of Combinatorial Theory* **38** (1985), 230-240.
- [5] D. Z. Du, F. K. Hwang, The Steiner ratio conjecture of Gilbert and Pollak is true, *Proc. Natl Acad. Sci. USA* **87** (1990), 9464-9466.
- [6] D. Dreyer, M. Overton, Two Heuristics for the Euclidean Steiner Tree Problem, *Journal of Global Optimization* **13** (2002), 95-106.
- [7] J. Van Laarhoven, Exact and heuristic algorithms for the Euclidean Steiner tree problem, dissertation, University of Iowa, (2010).