

# A Specialized Case of the Euclidean 3-Dimensional Steiner Tree Problem

Jacob Shetler

---

## Introduction

In a Steiner Tree problem, one must connect a given set of vertices together in order to create a connected graph of the least possible weight. In our problem, we define the weight to be the cumulative Euclidean length of all of the edges. While this sounds very similar to Minimum Spanning Tree problems, this is not the case. In Steiner problems one is allowed to add intermediate vertices and edges into the graph along with the given vertices. These intermediate vertices, called Steiner points, are then used like hubs to shorten the total length of the graph. Many forms of Steiner Tree problems have been studied and solved in two dimensions, but much less has been accomplished and published in three dimensions. In this report, we will analyze and create a Steiner Tree for a specific set of initial vertices in three dimensions.

Much work has been accomplished for two dimensional Steiner problems throughout the last 200 years. In 1879, a German mathematician Karl Bopp presented a dissertation entitled "On the shortest connection system for four points," in which he articulately detailed the construction of all possible Steiner Trees for four given vertices in two dimensions. From his work and others before him, it became evident that all edges coming into any Steiner point must form angles that are all  $120^\circ$ . Looking from the 21<sup>st</sup> century, this conclusion seems an obvious one: in order to minimize distance between points, one should always choose the point that creates the least possible equivalent angles. Along with this conclusion came another that there can be a maximum of  $n-2$  Steiner points for any given set of  $n$  vertices. At the end of this dissertation, Bopp presents observations for the 3-dimensional Steiner problem. One of these observations is that the same  $120^\circ$  rule must apply to Steiner points in three dimensions. Another observation is that any Steiner point must reside in the same plane as its direct neighbors. A corollary of these conclusions is that a Steiner point may only have a maximum of 3 direct neighbors (Brazil et al. 16-20).

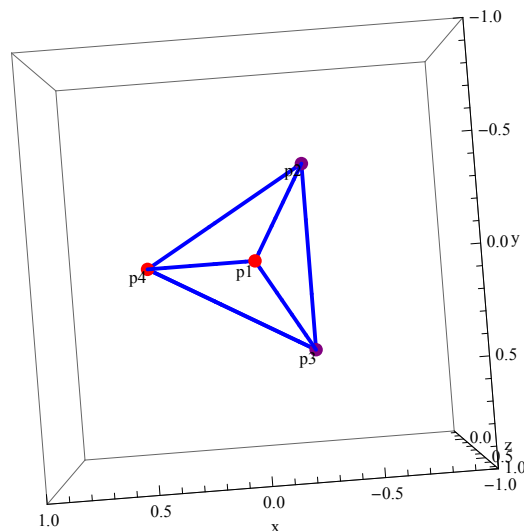
Continuing on from the work of Bopp and others we will look at creating Steiner trees for a set of vertices in three dimensions. The initial vertices will form the vertices of a regular tetrahedron. This tetrahedron will be centered at the origin and will edges of length 1. The four points and their 3D plot are shown below.

```

p1 = {0, 0,  $\sqrt{\frac{2}{3} - \frac{1}{2\sqrt{6}}}$ };
p2 = { $-\frac{1}{2\sqrt{3}}$ ,  $-\frac{1}{2}$ ,  $-\frac{1}{2\sqrt{6}}$ };
p3 = { $-\frac{1}{2\sqrt{3}}$ ,  $\frac{1}{2}$ ,  $-\frac{1}{2\sqrt{6}}$ };
p4 = { $\frac{1}{\sqrt{3}}$ , 0,  $-\frac{1}{2\sqrt{6}}$ };

points = ListPointPlot3D[{p1, p4}, PlotRange → {{-1, 1}, {-1, 1}, {-0.25, .75}},
  PlotStyle → Directive[Red, PointSize[Large]]];
points2 = ListPointPlot3D[{p2, p3}, PlotRange → {{-1, 1}, {-1, 1}, {-0.25, .75}},
  PlotStyle → Directive[Purple, PointSize[Large]]];
lines = Graphics3D[{Thick, Blue, Line[{p1, p2, p3, p4, p1, p3, p4, p2}]}];
labels = Graphics3D[{Text["p1", p1 + .05],
  Text["p2", p2 + .05], Text["p3", p3 + .05], Text["p4", p4 + .05]}];
Show[points, points2, lines, labels, AxesLabel → {"x", "y", "z"}, ViewPoint → Above]

```



If we were to use zero Steiner points, the total weight of the minimum spanning tree graph would be 3. By using Steiner points, we should be able to improve on this weight. We will look one possible solution for using one Steiner point and see if we can improve on that solution by adding another Steiner point.

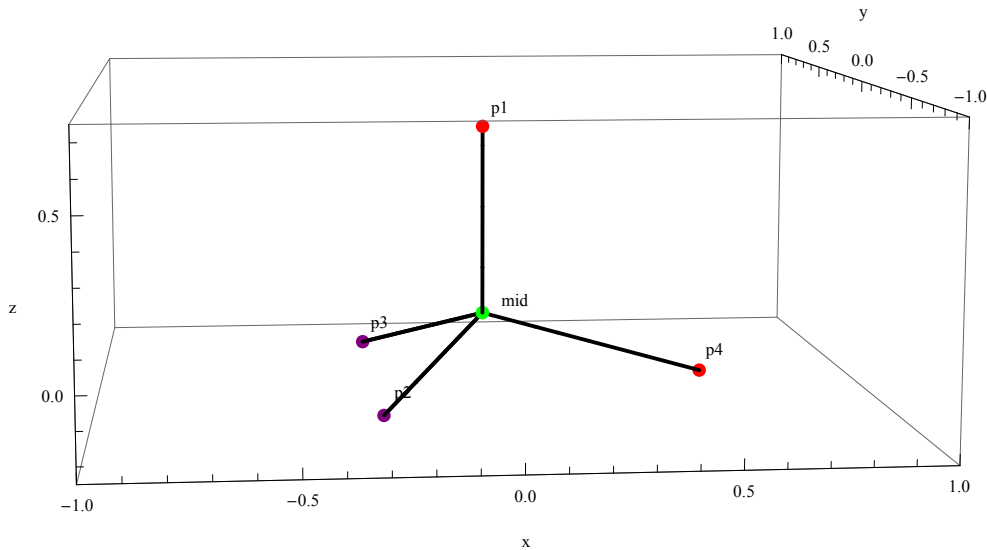
## One Steiner Point

With one Steiner point, we create what seems to be a very good solution to this problem. Similar to putting a single Steiner point at the Fermat point of a triangle in two dimensions, we place our single Steiner point at the center of the tetrahedron. We created our tetrahedron so its center was located at the origin, so this is fairly easy to do. The complete tree is shown below.

```

midPoint = {0, 0, 0};
midPointPlot = ListPointPlot3D[{midPoint},
  (*PlotRange→{- .25, .75}, *) PlotStyle → Directive[Green, PointSize[Large]]];
mLines = Graphics3D[{Thick, Black,
  Line[{p1, midPoint, p2, midPoint, p3, midPoint, p4}]}];
mLabel = Graphics3D[{Text["mid", midPoint + {.08, -.05, .05}]}];
Show[points, points2, mLines, midPointPlot,
  labels, mLabel, AxesLabel → {"x", "y", "z"}]

```



It is fairly easy to calculate the total weight of the graph. We will use *Mathematica* to calculate it for us.

```

oneSteinerWeight =
  EuclideanDistance[p1, midPoint] + EuclideanDistance[p2, midPoint] +
  EuclideanDistance[p3, midPoint] + EuclideanDistance[p4, midPoint];
N[oneSteinerWeight]
2.44949

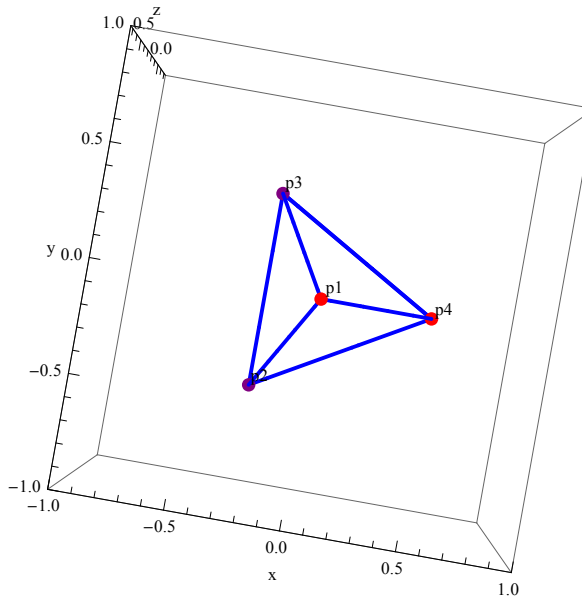
```

And the total weight of the graph with one Steiner point is about 2.44949. We now move on to calculate the location and points for this graph with two Steiner points.

## Two Steiner Points

We know from our introduction and previous published work that any Steiner points in three dimensions must connect to three other points at  $120^\circ$  angles. It seems fairly obvious what our mode of attack should be to find the minimum Steiner tree for the vertices of a tetrahedron with two Steiner points. The two Steiner points must exist within the tetrahedron at  $120^\circ$  angles to separate pairs of two vertices. The other point that the Steiner points will connect to will be the other Steiner point. The first step is to pair the vertices into two groups and then find all points that are  $120^\circ$  away from both of the points. Because we are in 3-dimensional space, the set of these vertices will form a circle.

```
Show[points, points2, lines, labels, AxesLabel -> {"x", "y", "z"}, ViewPoint -> Above]
```



We separate the points into two groups:  $\{p2, p3\}$  and  $\{p1, p4\}$ . We can arbitrarily separate them into groups because the figure is completely symmetrical. We have color-coated them in the graph in order to make the distinction clear.

Our first (and easier) goal will be to find the circle of points that form an interior angle of  $120^\circ$  with both  $p2$  and  $p3$ . This is fairly simple because this circle exists in the  $x$ - $z$  plane. We know that the center of the circle is the midpoint between  $p2$  and  $p3$ . Furthermore, we can find the radius of the circle by utilizing the relationship between the sides of a 30-60-90 triangle. Since the radius will be measured straight out from the midpoint of  $p2$  and  $p3$ , we are trying to find the length of the side opposite of the  $30^\circ$  angle. Since the length of the side opposite of the  $60^\circ$  angle is  $\frac{1}{2}$ , the length of the side opposite the  $30^\circ$  angle must be  $\frac{1}{2\sqrt{3}}$ . Armed with our center and radius, we plug these into the equation of a circle to find all of the points for the  $\{p2, p3\}$  circle.

```
centerPointS1 = Mean[{p2, p3}];
```

```
radiusS1 =  $\frac{1}{2\sqrt{3}}$ ;
```

```
circleS1 = (x - centerPointS1[[1]])^2 + (z - centerPointS1[[3]])^2 == (radiusS1)^2
```

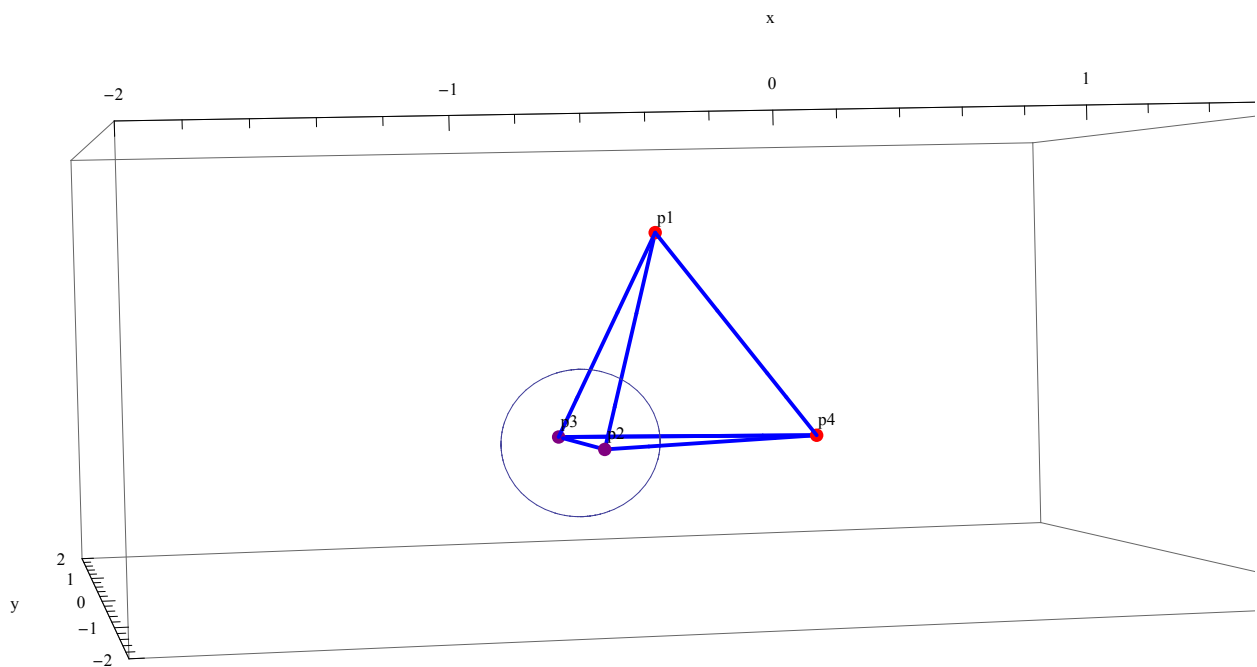
$$\left(\frac{1}{2\sqrt{3}} + x\right)^2 + \left(\frac{1}{2\sqrt{6}} + z\right)^2 = \frac{1}{12}$$

And we can plot this circle (now called circleS1) to give a better representation of what we just calculated.

```

points = ListPointPlot3D[{p1, p4}, PlotRange → {{-2, 2}, {-2, 2}, {- .75, 1}},
  PlotStyle → Directive[Red, PointSize[Large]]];
points2 = ListPointPlot3D[{p2, p3}, PlotRange → {{-2, 2}, {-2, 2}, {- .75, 1}},
  PlotStyle → Directive[Purple, PointSize[Large]]];
circleAroundS1 = ParametricPlot3D[{{  $\frac{1}{2\sqrt{3}} \cos[\theta] - \frac{1}{2\sqrt{3}}$ ,
  0,  $\frac{1}{2\sqrt{3}} \sin[\theta] - \frac{1}{2\sqrt{6}}$  }, { $\theta$ , 0, 2  $\pi$ }}];
Show[points, points2, lines, labels, circleAroundS1, AxesLabel → {"x", "y", "z"}]

```



We can conclude from our calculations and earlier conclusions that the Steiner point connecting p2, p3, and the other Steiner point must lie on the circle  $\left(\frac{1}{2\sqrt{3}} + x\right)^2 + \left(\frac{1}{2\sqrt{6}} + z\right)^2 = \frac{1}{12}$ .

Our next goal, and a much more difficult one, will be to find the circle of points around p1 and p4 that satisfy the same properties circleS1 did for p2 and p3.

We know that the midpoint of this circle must be halfway between p1 and p4. We also know that the radius of this circle will be the same as circleS1. We are not done at this point because this circle is tilted in our 3-dimensional space: it does not lie on any of the planes created by two of our three axes. Nevertheless, we can use orthogonal matrices to rotate a circle in the x-y plane about the origin in order to find the equation of this circle.

Our first step is to find the equation of this circle (now called circleS2) in the x-y plane. This equation is

$$\text{centerPointS2} = \text{Mean}[\{\mathbf{p1}, \mathbf{p4}\}];$$

$$\text{radiusXYS2} = \frac{1}{2\sqrt{3}};$$

$$\text{circleXYS2} = (\mathbf{x} - \text{centerPointS2}[[1]])^2 + (\mathbf{y} - \text{centerPointS2}[[2]])^2 == (\text{radiusXYS2})^2$$

$$\left(-\frac{1}{2\sqrt{3}} + \mathbf{x}\right)^2 + \mathbf{y}^2 == \frac{1}{12}$$

We know need to find the equation of the plane on which that the transformed circle will lie. This is also fairly simple to do, because we know that this plane must also pass through p2 and p3. We know this because the requirement that all points that Steiner points connect to must be in the same plane and thus the Steiner point on circleS2 must be coplanar to p2 and p3. We have a plane passing through three points and thus the plane is given by the equation

$$\text{Simplify}\left[\mathbf{x}^2 + \mathbf{y}^2 + \left(\mathbf{z} - \frac{\sqrt{\frac{3}{2}}}{2}\right)^2 == \left(\mathbf{x} - \frac{1}{\sqrt{3}}\right)^2 + \mathbf{y}^2 + \left(\mathbf{z} + \frac{1}{2\sqrt{6}}\right)^2\right]$$

$$\mathbf{x} == \sqrt{2} \mathbf{z}$$

$\mathbf{x} = \sqrt{2} \mathbf{z}$ . We now need 3 orthogonal vectors in order to create our orthogonal matrix. Two of them will exist in this plane; the other will be orthogonal to the plane. We know that this orthogonal vector will be the same as the direction vector between p1 and p4, so we calculate it and use a scalar multiple of it in order to make it a little more readable.

$$\mathbf{v3} = \mathbf{p4} - \mathbf{p1};$$

$$\mathbf{v3} = \mathbf{v3} * \sqrt{3}$$

$$\{1, 0, -\sqrt{2}\}$$

We now must calculate two other vectors in the  $\mathbf{x} = \sqrt{2} \mathbf{z}$  plane. One of these is fairly easy to calculate, we just make it the “opposite” of v3.

$$\mathbf{v1} = \{\sqrt{2}, 0, 1\}$$

$$\{\sqrt{2}, 0, 1\}$$

For our last vector, we take the cross-product of v1 and v3. Any scalar multiple of this vector will also be orthogonal, so we multiply it by  $\frac{1}{3}$  for readability.

$$\mathbf{v2} = \text{Cross}[\mathbf{v1}, \mathbf{v3}] * \frac{1}{3}$$

$$\{0, 1, 0\}$$

We now need to normalize our vectors and use them in columns of our orthogonal transformation rotation matrix, called A.

```

v1Norm = Normalize[v1]; v2Norm = Normalize[v2]; v3Norm = Normalize[v3];
A = {{v1Norm[[1]], v2Norm[[1]], v3Norm[[1]]},
     {v1Norm[[2]], v2Norm[[2]], v3Norm[[2]]}, {v1Norm[[3]], v2Norm[[3]], v3Norm[[3]]}};
A // MatrixForm

```

$$\begin{pmatrix} \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{3}} & 0 & -\sqrt{\frac{2}{3}} \end{pmatrix}$$

It then becomes easy to generate our new coordinate system to rotate our circle.

```

newCoords = A.{{x}, {y}, {z}};
newCoords // MatrixForm

```

$$\begin{pmatrix} \sqrt{\frac{2}{3}} x + \frac{z}{\sqrt{3}} \\ y \\ \frac{x}{\sqrt{3}} - \sqrt{\frac{2}{3}} z \end{pmatrix}$$

By substituting these new coordinates into our original equation of the circle, we find our real equation of the circle. Note that we have shifted our center from  $(0, \frac{1}{2\sqrt{3}})$  to  $(0, \frac{1}{2\sqrt{2}})$ . This is to account for rotating about the origin instead of about the center of our s2 circle. The new center was chosen so that the rotation puts the center back in the correct location. We now substitute our variables into the original equation  $(-\frac{1}{2\sqrt{2}} + x)^2 + y^2 = \frac{1}{12}$  to find the equation of circleS2. Note that the original equation does not include any z's, nor does y require any change of variable.

$$\text{circleS2} = \left(-\frac{1}{2\sqrt{2}} + x\right)^2 + y^2 = \frac{1}{12} \quad / . \quad x \rightarrow \sqrt{\frac{2}{3}} x + \frac{z}{\sqrt{3}}$$

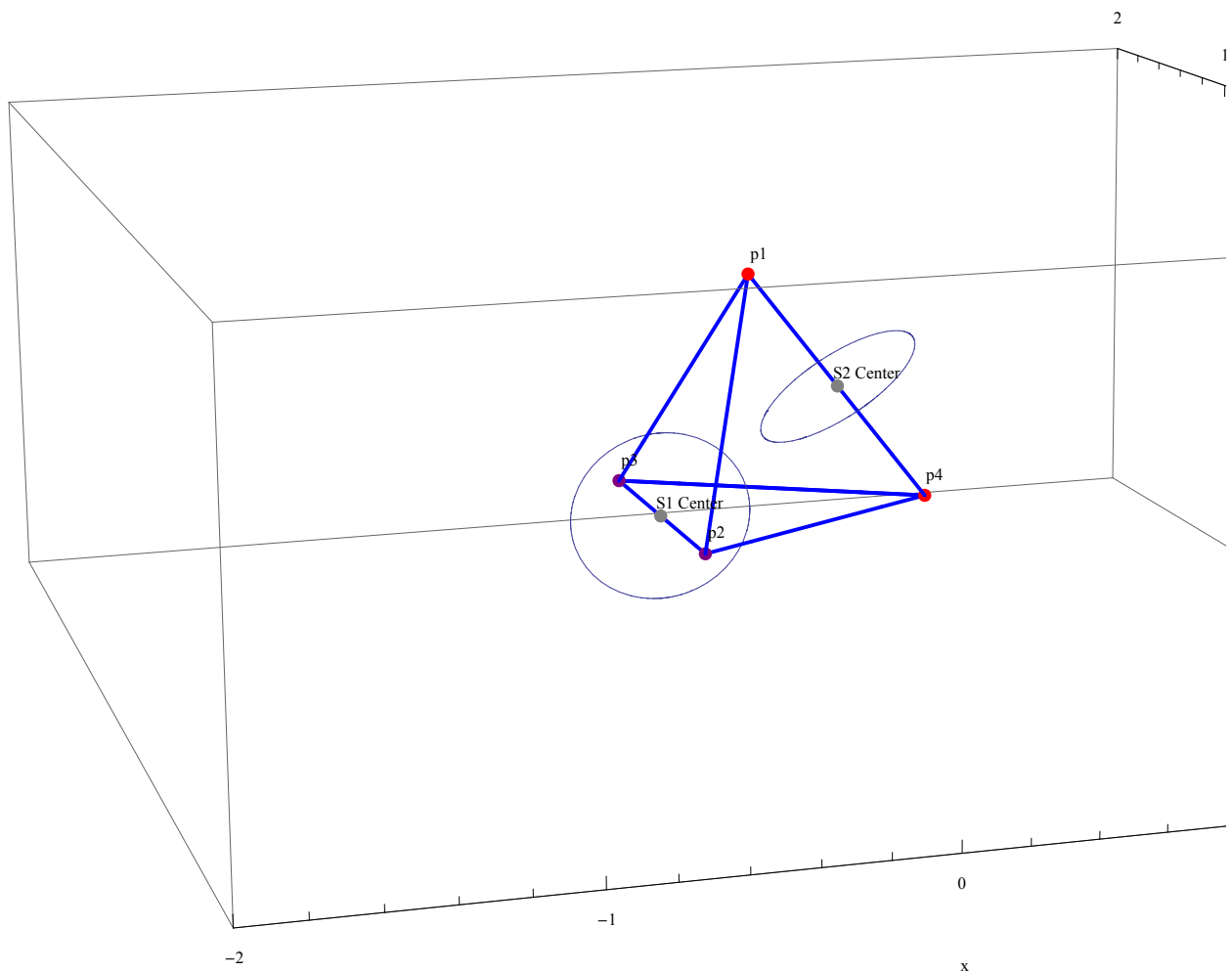
$$y^2 + \left(-\frac{1}{2\sqrt{2}} + \sqrt{\frac{2}{3}} x + \frac{z}{\sqrt{3}}\right)^2 = \frac{1}{12}$$

We can now plot this circle with everything else.

```

circleArounds2 =
  ParametricPlot3D[A. {  $\frac{1}{2\sqrt{3}} \cos[\theta] + \frac{1}{\sqrt{8}}$ ,  $\frac{1}{2\sqrt{3}} \sin[\theta]$ , 0 }, { $\theta$ , 0, 2  $\pi$  }];
points = ListPointPlot3D[{p1, p4}, PlotRange -> {{-2, 2}, {-2, 2}, {- .75, 1}},
  PlotStyle -> Directive[Red, PointSize[Large]]];
points2 = ListPointPlot3D[{p2, p3}, PlotRange -> {{-2, 2}, {-2, 2}, {- .75, 1}},
  PlotStyle -> Directive[Purple, PointSize[Large]]];
circleCenters = ListPointPlot3D[{centerPointS1, centerPointS2}, PlotRange ->
  {{-2, 2}, {-2, 2}, {- .75, 1}}, PlotStyle -> Directive[Gray, PointSize[Large]]];
circleCenterLabels = Graphics3D[{Text["S1 Center", centerPointS1 + {.08, -.05, .05}],
  Text["S2 Center", centerPointS2 + {.08, -.05, .05}]}];
Show[points, points2, lines, labels, circleArounds1, circleArounds2,
  circleCenters, circleCenterLabels, AxesLabel -> {"x", "y", "z"}]

```



Our problem now becomes one of finding the points on the circles that are closest to one another. These two points will thus become our two Steiner points, with a minimum distance between them. This will in turn minimize the total weight of our Steiner Tree solution. Luckily, this ends up being much easier than one might expect at first glance, because the points closest to each other must lie on the line segment connecting the two centers. If this is not immediately clear to you, consider the following argument.



If we were to fix one of the points on circleS1 as SteinerPointS1, then in order to minimize the distance from SteinerPointS1 to circleS2 we must create a line that passes through SteinerPointS1 and is normal to the arc of circleS2 that it intersects. This implies that this line continues through the center of circleS2. But this argument can also be used if you were to first fix the point on circleS2. Thus the line connecting the two Steiner points must go through the centers of circleS1 and circleS2.

The line connecting the two centers is given by

```
centerS1 = Mean[{p2, p3}];
centerS2 = Simplify[Mean[{p1, p4}]];
2 * (x - centerPointS1[[1]]) / (centerS2[[1]] - centerS1[[1]]) ==
  - (z - centerPointS1[[3]]) / (centerS2[[3]] - centerS1[[3]])
```

$$2\sqrt{3} \left( \frac{1}{2\sqrt{3}} + x \right) = \sqrt{6} \left( -\frac{1}{2\sqrt{6}} - z \right)$$

in general form. If we wish to graph it in parametric form, we have the equations

$$x = \frac{1}{\sqrt{3}} t$$

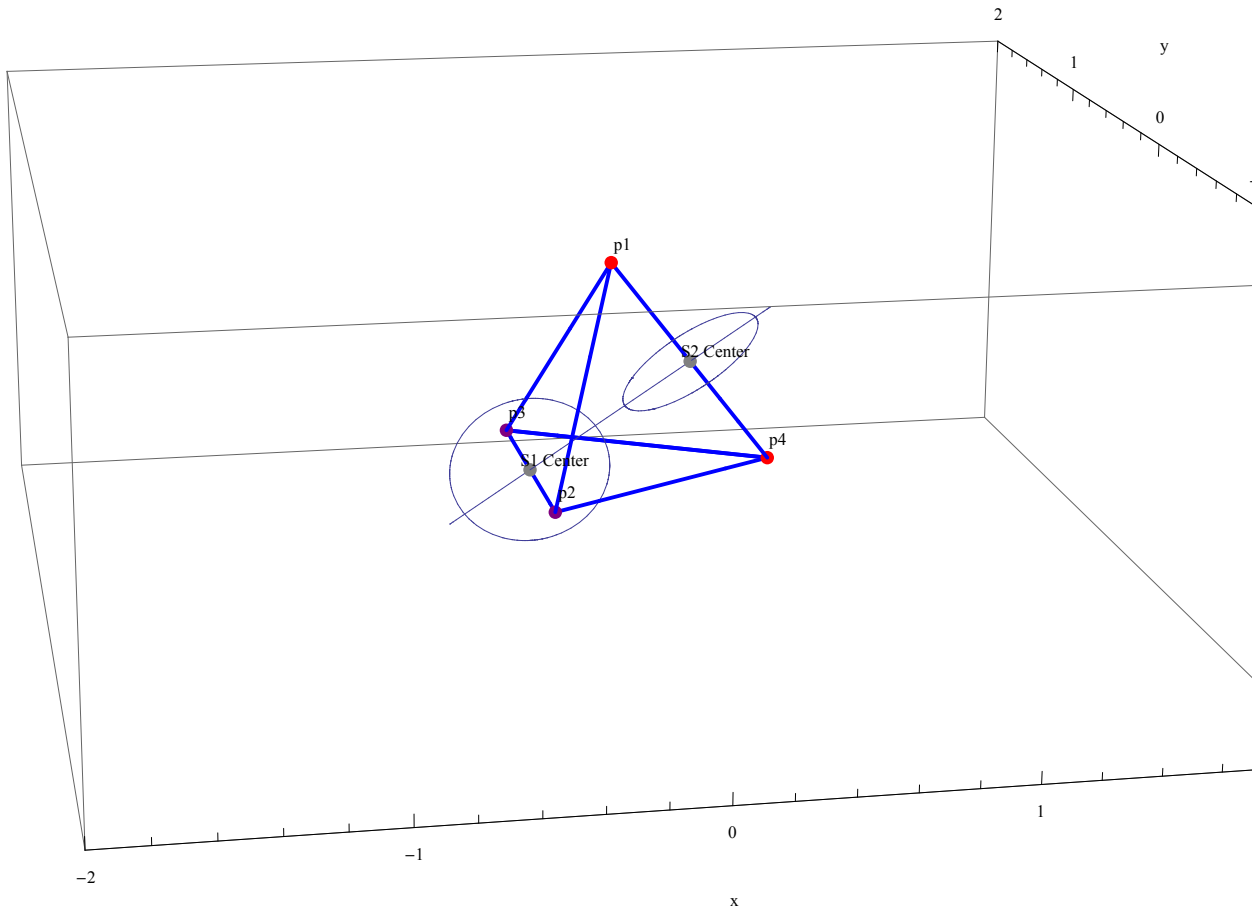
$$y = 0$$

$$z = \frac{1}{\sqrt{6}} t$$

```

centerLine = ParametricPlot3D[{{ $\frac{1}{\sqrt{3}}$  t, 0,  $\frac{1}{\sqrt{6}}$  t}, {t, -1, 1}];
Show[points, points2, lines, labels, centerLine, circleAroundS1,
circleAroundS2, circleCenters, circleCenterLabels, AxesLabel -> {"x", "y", "z"}]

```



And it becomes fairly trivial to determine where this line intersects our two circles. Because our entire figure is symmetric, we are able to simply find the correct intersection with circleS1 and negate the coordinates for the intersection of circleS2.

The equation of the line is

$$z = \frac{1}{\sqrt{2}} x$$

and the equation of the circle is

$$\frac{1}{12} = \left(x + \frac{1}{2\sqrt{3}}\right)^2 + \left(z + \frac{1}{2\sqrt{6}}\right)^2.$$

Plugging in  $\frac{x}{\sqrt{2}}$  for z in the circle equation, solving, and resolving for  $z = \frac{x}{\sqrt{2}}$  gives us two possible

points:

$$\left\{ \frac{1}{6} \left(-\sqrt{2} - \sqrt{3}\right), 0, \frac{-\sqrt{2} - \sqrt{3}}{6\sqrt{2}} \right\}$$

and

$$\left\{ \frac{1}{6} \left(\sqrt{2} - \sqrt{3}\right), 0, \frac{\sqrt{2} - \sqrt{3}}{6\sqrt{2}} \right\}.$$

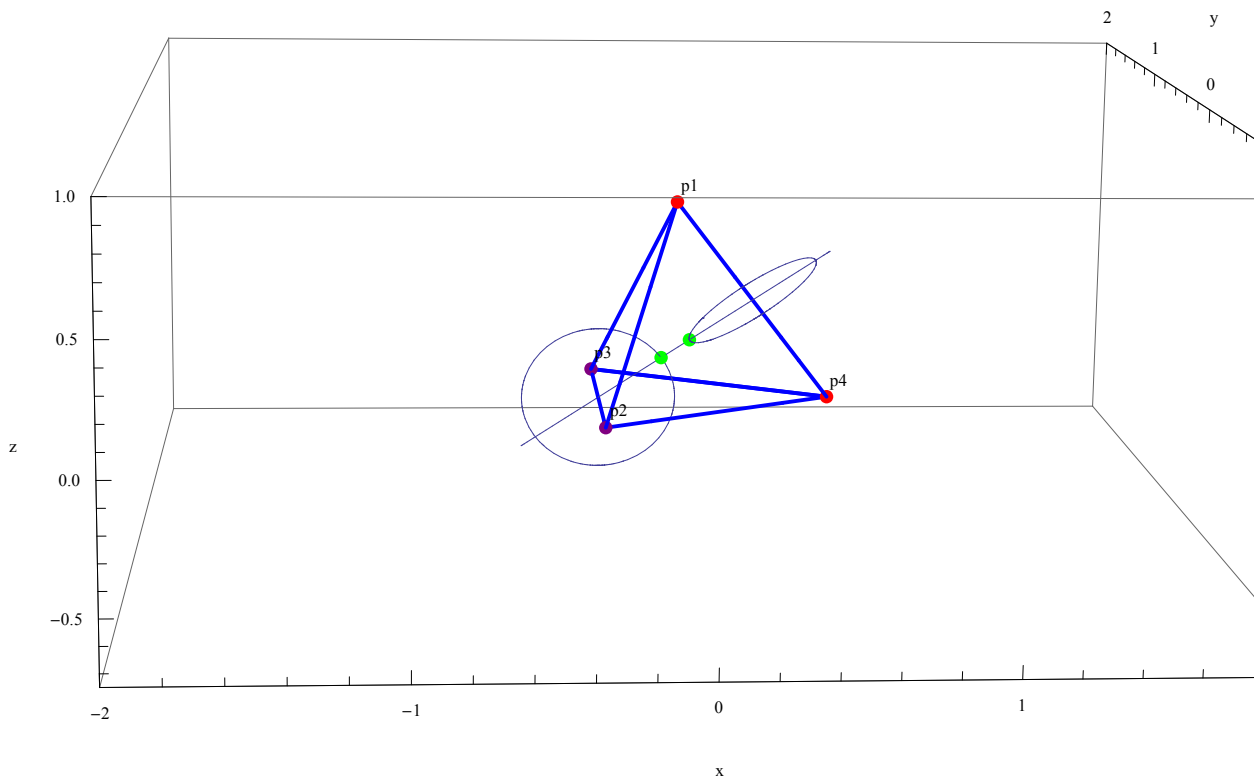
We obviously want the second point. We can negate it to find the other intersection. We first graph the intersections on the circles to give a better representation of what we just discovered.

$$\text{intersection2} = \left\{ \frac{1}{6} (\sqrt{2} - \sqrt{3}), 0, \frac{\sqrt{2} - \sqrt{3}}{6\sqrt{2}} \right\};$$

```

intersectionS1Point = intersection2;
intersectionS2Point = -intersection2;
intersectionS1 = ListPointPlot3D[{intersectionS1Point},
  PlotStyle → Directive[Green, PointSize[Large]]];
intersectionS2 = ListPointPlot3D[{intersectionS2Point},
  PlotStyle → Directive[Green, PointSize[Large]]];
Show[points, points2, lines, labels, centerLine, intersectionS1, intersectionS2,
  (*planeForS2,*) circleAroundS1, circleAroundS2, AxesLabel → {"x", "y", "z"}]

```

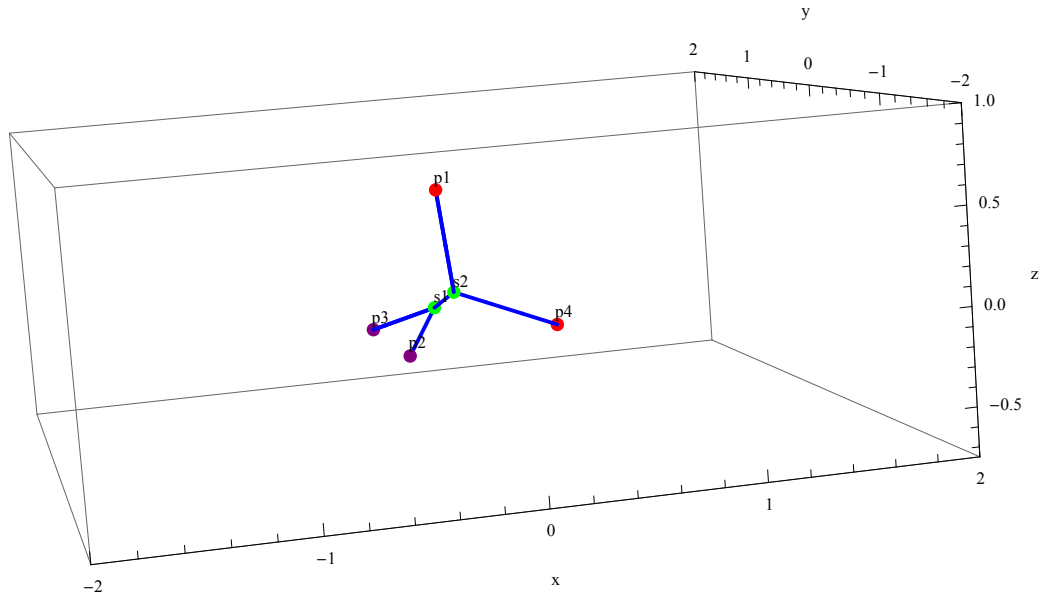


Lastly, we graph our Steiner solution for this Steiner tree problem using two Steiner points and calculate the total weight of this solution.

```

steinerLines =
  Graphics3D[{Thick, Blue, Line[{p2, intersectionS1Point, p3, intersectionS1Point,
    intersectionS2Point, p1, intersectionS2Point, p4}]}];
steinerLabels = Graphics3D[{Text["s1", intersectionS1Point + .05],
  Text["s2", intersectionS2Point + .05]}];
Show[points, points2, steinerLines, intersectionS1, intersectionS2,
  labels, steinerLabels, AxesLabel -> {"x", "y", "z"}]

```



```

twoSteinerWeight = EuclideanDistance[intersectionS1Point, intersectionS2Point] +
  EuclideanDistance[p1, intersectionS2Point] +
  EuclideanDistance[p4, intersectionS2Point] +
  EuclideanDistance[p2, intersectionS1Point] +
  EuclideanDistance[p3, intersectionS1Point];
N[twoSteinerWeight]
2.43916

```

And this weight is about 2.43916.

## Conclusions

In this analysis, we have generated minimum Steiner Trees using one and two Steiner points for an initial set of vertices that create a regular tetrahedron. The weight for the solution with one Steiner point was about 2.44949, and the solution with two points was about 2.43916. Thus we conclude that using two Steiner point is better than using one: this is what we had expected at the outset. We also see that using one or two Steiner points is better than using none, which gave us a weight of 3. It is important to note that using two points uses about 99.6% of the weight used by the solution with one Steiner point, so they are fairly similar in their final weights.

This conclusion is important to remember for practical applications of Steiner tree problems. Steiner tree problems have long been associated with solving problems of network design. For example, a network engineer may want to save costs by laying as little wire as possible between corporations

in different cities when creating a WAN. If the cities were set up in a regular tetrahedron (however unlikely this may be), our solution would provide an applicable design for laying the minimum amount of cable between the cities. Furthermore, the engineer could determine if saving 0.4% on the length of wire by adding an extra hub to the network would be worth the cost of buying that extra hub. In this way, our solution provides one basic step in a long process of determining the best network design for more complicated network systems.

Finally, we conclude with some drawbacks of our approach and how progress can be made on our solution in the future. One drawback is that we assumed that the maximum number of Steiner points in 3-dimensional space is the same as the maximum number of Steiner points in two dimensions. We did not consider how adding a third Steiner point would affect the result, or whether adding a third Steiner point makes sense at all. Another drawback is that we assumed that the solution provided for using one Steiner point is the optimal solution. We also did not analyze other solutions utilizing only one Steiner point. In the future, we hope that more progress will be made and published about Steiner tree problems in three dimensions. We also see possibilities of further analysis regarding Steiner tree problems with initial vertices that form other regular polyhedra. These further studies and solutions will provide network and circuit designers with better optimal designs, potentially reducing wasted resources throughout the world.

---

## References

### References and Resources Used

Brazil, Marcus, Ronald L. Graham, Doreen A. Thomas, and Martin Zachariasen. "On the History of the Euclidean Steiner Tree Problem." *Archive for History of Exact Sciences*. (August 2013):15-20. Web.

*Mathematica* Version 9.0.1.0

Dr. David Housman, Goshen College Mathematics Department