1997

Steiner Trees Over Generalized Checkerboards

Meta M. Voelker '97
Illinois Wesleyan University

STEINER TREES OVER GENERALIZED CHECKERBOARDS

BY

META M. VOELKER

THESIS

Submitted in partial fulfillment of the requirements
for Honors Research in Mathematics
at Illinois Wesleyan University, 1997

Bloomington, Illinois

STEINER TREES OVER GENERALIZED CHECKERBOARDS

Meta M. Voelker
Department of Mathematics
Illinois Wesleyan University, 1997
Robin Sue Sanders, Advisor

To minimize the length of a planar network, we can build a Steiner minimal tree — that is, a tree consisting of the original network points, as well as additional, strategically-placed (Steiner) points. Chung, Gardner and Graham [2] investigated building Steiner trees over grids of unit squares. We generalize their ideas to grids of rhombuses, and show that two near-optimal Steiner trees exist for each grid, one built from Steiner trees over rhombuses and one built from Steiner trees over isosceles triangles. Further, we conjecture that for grids with an odd number of layers, only the small angle of the rhombus drives which tree is shorter; for grids with an even number of layers, the small angle is the most important factor in determining which scheme to use.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Networks of all kinds are encountered everywhere, and often the cost of building these networks involves the overall length of the network. Minimizing the overall length is thus advantageous. This leads us to the following problem:

Problem 1 (Jarnik and Kössler Problem [15]) *Find the shortest network spanning n points in the plane.*

To solve this problem using only the given network points, we build what is known as a *minimum spanning tree* (see Section 2.1). Essentially, this is accomplished by finding the shortest connections between the n network points.

However, the minimum spanning tree does not always return the absolute shortest network. If we add additional, strategically-placed points to the network, we can actually shorten the distances between the network points and thus shorten the overall length of the network. These additional points are referred to as *Steiner points*, and the corresponding network is referred to as a *Steiner tree*. The problem of finding the placement and the number of Steiner points such that the resulting network has minimum length is referred to as the *Steiner Problem*.

According to [15], the origin of the Steiner Problem has very little to do with Jacob Steiner (1796–1863), the mathematician for whom the problem is named. Steiner did work on the problem in one of its variant forms, but the problem itself is actually based on two similar problems, the Jarnik and Kössler Problem (stated in Problem 1) and the Generalized Fermat Problem, stated below.

Problem 2 (Generalized Fermat Problem [15]) *Find the point in the plane that minimizes the distances from itself to n given points.*

It was not until Courant and Robbins' *What is Mathematics?* [4] that the problem became known as the Steiner Problem [15].

The solution to the Steiner Problem has numerous applications, from the construction of roadways and (computer) networks to building evolutionary trees in biology [8]. Unfortunately, though, the problem has been shown to be NP-complete [10, 11], implying that no efficient algorithm exists (and may never exist) to solve the general case. However, by restricting ourselves to special network configurations, we can often find (at least) near-optimal solutions. One such example is Chung, Gardner and Graham's work on planar grids of squares [2, 3]. In their work, they were able to build near-optimal networks (some of which have been proven to be optimal) over such grids. One of their conjectured networks is shown in Figure 1.1.

In this paper we generalize Chung, Gardner and Graham's work to rhombuses. That is, we build near-optimal networks over points that are arranged in grids made of rhombuses, otherwise referred to as *generalized checkerboards.*

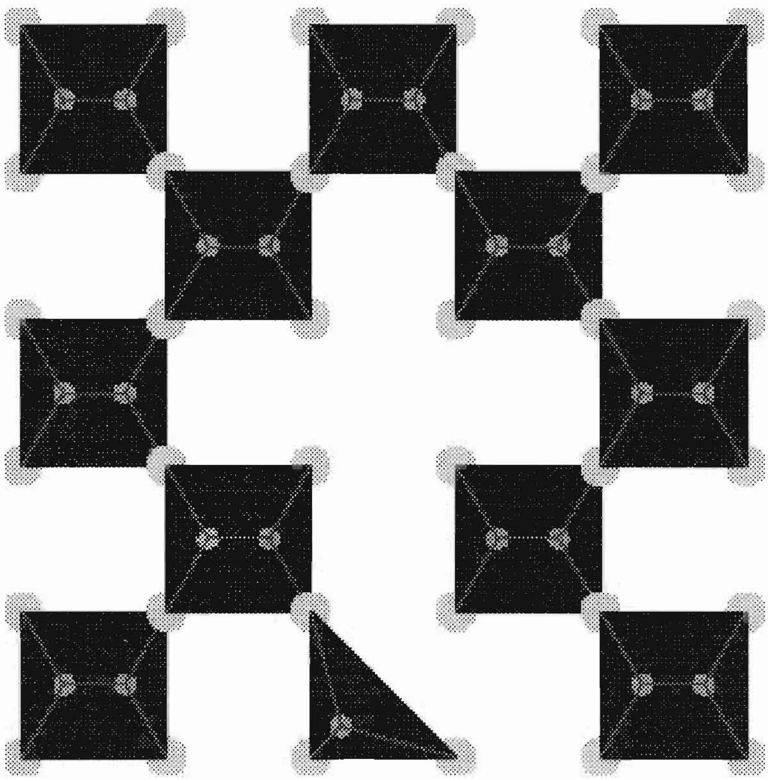


Figure 1.1 The conjectured Steiner minimal tree over the 5×5 checkerboard.

CHAPTER 2

DEFINITION OF TERMS AND KNOWN RESULTS

2.1 Definition of terms

In this section, we define the terms from graph theory used throughout this paper. Additional information can be found in [19].

Definition 1 Network (Plane Graph) : *A set of points (vertices) and lines (edges). The lines must be straight and can only intersect at points.*

Definition 2 Network Point : *One of the original n points; a point that has* not *been added to the network. Throughout this paper we will use capital letters to denote network points.*

Definition 3 Tree : *A network that connects the points in a manner such that there is exactly one path between any pair of distinct points in the network, i.e. there are no cycles or circuits in the network.*

Definition 4 Minimum Spanning Tree : *A tree that spans the network points in such a way as to minimize the length of the network. Only the network points are contained in the network's minimum spanning tree.*

Definition 5 Steiner Tree : *A tree that allows additional, strategically-placed points to be added to the network in order to reduce the length of the minimum spanning tree further.*

Definition 6 Steiner Minimal Tree : *A Steiner tree that attains minimum length.*

Definition 7 Steiner Point : *A point that has been added to the network in order to shorten the length of the network. Throughout this paper we will use S and S_i's to denote Steiner points.*

2.2 Properties of Steiner minimal trees

In this section, we summarize the properties of Steiner trees we will use in this paper. Additional properties can be found in [12].

Theorem 1 *A Steiner point is the junction of 3 lines.*

Theorem 2 *The three lines meeting at a Steiner point meet at angles of* $120°$.

In order to prove Theorems 1 and 2, we make use of the following lemma.

Lemma 1 *In a Steiner minimal tree, no pair of lines meet at less than* $120°$.

Proof for Lemma 1 (from [12]): Let the given Steiner minimal tree be interpreted as a mechanical system in which potential energy is the sum of the distances between adjacent points. Then the Steiner tree is in stable equilibrium when the tree attains minimum length. Now, assume by way of contradiction, that two lines of the Steiner tree, say lines PR and RQ meet with $\angle PRQ = \theta$ where $\theta < 120°$ (see Figure 2.1). Using

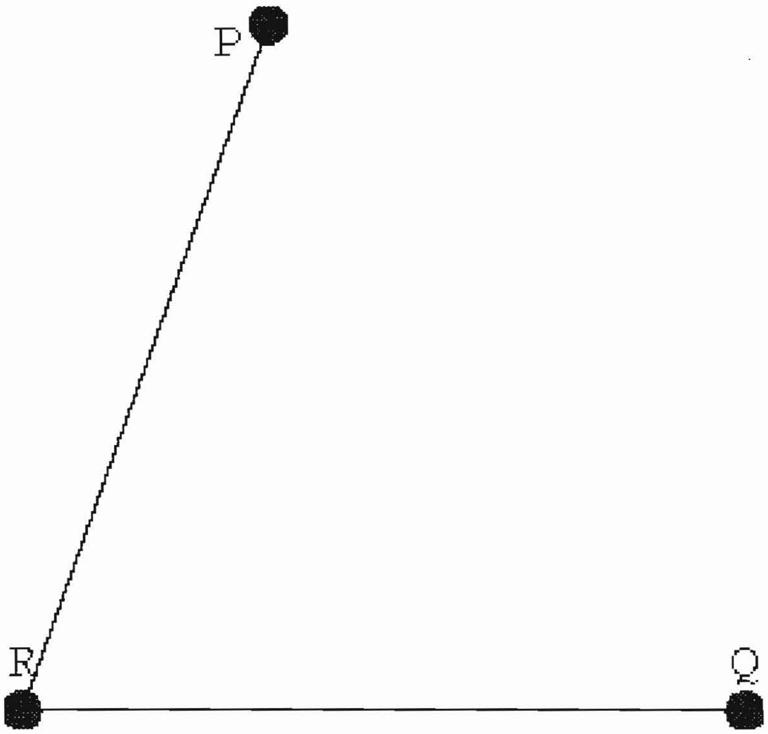


Figure 2.1 Lines PR and RQ meet with $\angle PRQ = \theta < 120°$.

the mechanical interpretation, these two lines pull on point R with resultant force of magnitude $F = 2\cos(\theta/2) > 1$. Now consider the effect of splitting R by adding a Steiner point S at R and replacing lines PR and RQ by PS, QS and RS. The unit force of RS is inadequate to hold S at R against the combined force F exerted by QS and RS. Thus S is pulled away from R and we obtain a configuration with a lower potential energy and a shorter length (Figure 2.2), a contradiction. □

The proof for Theorems 1 and 2 now follow: Lemma 1 implies a Steiner minimal tree can have no point incident to more than three lines. Since a Steiner point must be incident to at least three lines (otherwise no reduction in total tree length occurs), every Steiner point is incident to three lines that meet at angles of 120°.

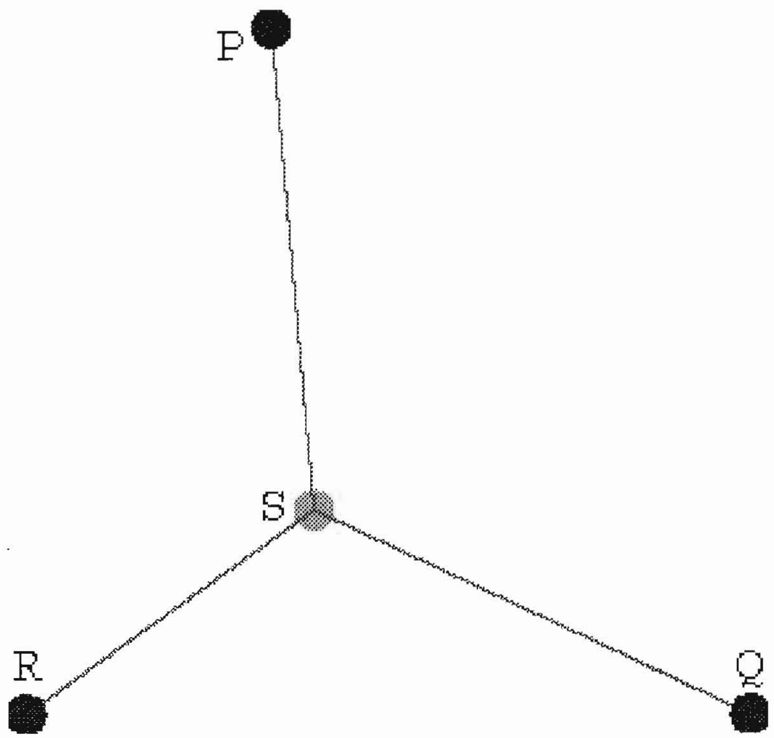


Figure 2.2 Point S is pulled away from R.

Theorem 3 *The total number of Steiner points in a Steiner tree is at most $n-2$, where n is the total number of network points.*

Proof (from [12]): From graph theory, we know every tree has one more point than it has lines. Thus a tree with n network points and s Steiner points must have $n+s-1$ total lines. Since each line has two ends, the number of incident lines obtained by summing over all the points is $2(n+s-1)$. Now, if n_k of the network points have k incident lines, then we have

$$2(n+s-1) = 3s + \sum_k kn_k.$$

Since $n = \sum_k n_k$, we have

$$2s - 2 + 2\sum_k n_k = 3s + \sum_k kn_k$$

which implies

$$s = -2 + \sum_k (2-k)n_k = n_1 - 2 - n_3 - 2n_4 - \ldots.$$

In particular, $s \leq n-2$ with equality holding if and only if each network point is incident to only one line. □

Steiner trees with exactly $n-2$ Steiner points are referred to as *full* Steiner trees. In a full Steiner tree all network points are incident to only one line of the Steiner tree.

Theorem 4 *All Steiner trees are either full Steiner trees or can be decomposed into a union of full Steiner trees.*

Proof (from [12]): Consider a given Steiner tree. If this tree is full we are done, so assume the tree is not full. Because it is not full, there exists at least one network point that is incident to more than one line. We then can break the original Steiner tree into full components as follows:

1. Replace each network point A_i that is incident to k (for $k > 1$) lines with disconnected points $A_{i,1}, \ldots, A_{i,k}$, all located at point A_i.

2. Connect each of the k lines that were incident at A_i to one of the newly-created points $A_{i,1}, \ldots, A_{i,k}$. (Each of the points $A_{i,1}, \ldots, A_{i,k}$ is now incident to only one line.)

This gives us several smaller full Steiner trees whose union forms the original Steiner tree. □

When building a Steiner tree, it is useful to know where a Steiner point can and cannot be placed. This leads us to the following definition:

Definition 8 Steiner Hull : *A set of points from the plane that contains both the network points and the Steiner points of the given Steiner minimal tree.*

The plane itself is a trivial example of a Steiner hull for any planar Steiner minimal tree. However, we can find smaller Steiner hulls.

Theorem 5 *The convex hull over the set of network points forms a Steiner hull for any Steiner minimal tree connecting the network points.*

Proof: Recall that the convex hull over the set of network points spans the network points in such a way as to contain entirely every line segment joining any two points in

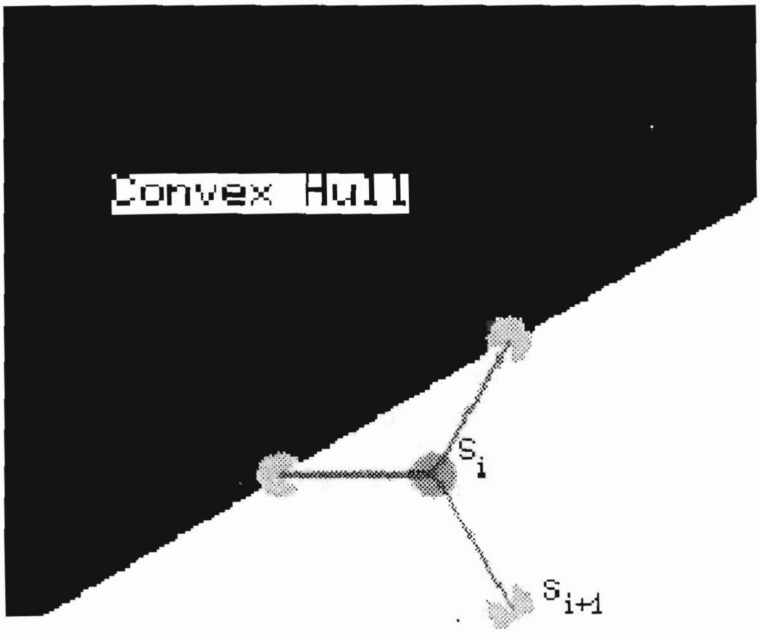


Figure 2.3 One line leaving S_i points away from the convex hull.

the hull. Now, assume by way of contradiction that the convex hull is not a Steiner hull. Then there exists some Steiner point, say S_i, of the Steiner minimal tree that lies outside of the convex hull. Theorems 1 and 2 imply at least one of the lines leaving S_i points away from the convex hull (see Figure 2.3). Since all of the network points lie inside the convex hull by definition, S_i must be connected to some other Steiner point, say S_{i+1}, that also lies outside of the convex hull. Using the same argument on S_{i+1}, we can find S_{i+2} that lies outside the convex hull. Continuing in this manner, we can generate an infinite series of Steiner points that lie outside the convex hull, thus implying that the resulting Steiner tree cannot be minimal, a contradiction. □

For alternate proofs to Theorem 5, see [12, 15]. This result is useful in that it defines a particular region in which to look for Steiner points: we never look outside of the convex

hull. Thus, we will be concentrating on finding Steiner points located in the "interior" of generalized checkerboards.

2.3 Connections between minimum spanning trees and Steiner minimal trees

It is natural to ask whether there are connections between minimum spanning trees and Steiner minimal trees. Both trees work to minimize the overall length of the network, with differences. Minimum spanning trees use only the network points, while Steiner minimal trees add additional points to the network. Although minimum spanning trees are longer than Steiner minimal trees, minimum spanning trees are easier to build. Unlike Steiner minimal trees, which can be difficult to build in general, minimum spanning trees are *always* easily built by finding the shortest connections between the network points (see Kruskal's greedy algorithm in [19]).

Besides being easier to build, a network's minimum spanning tree may also be helpful in building good Steiner trees for that network and in testing these trees for minimality.

Theorem 6 *If a Steiner minimal tree contains a line that connects two network points, then this line must also be a line of a minimum spanning tree for the network.*

Proof (from [12]): Let A_1A_2 be a line in a Steiner minimal tree that connects the network points A_1 and A_2. Then all the points of the tree can be placed into one of two sets, C_1 or C_2, where C_1 contains all the points (both network and Steiner) that can be reached from A_1 without first passing through A_2 and C_2 contains all the points (again, both network and Steiner) that can be reached from A_2 without first passing through A_1

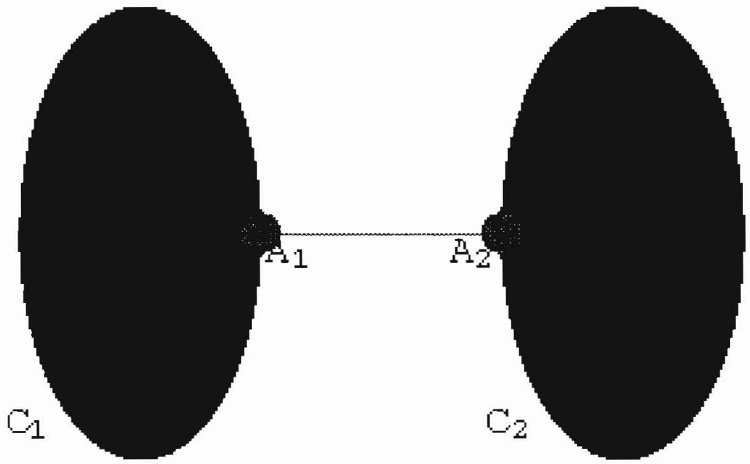


Figure 2.4 All the points of the tree can be placed into either C_1 or C_2.

(see Figure 2.4). The line A_1A_2 must be a connection between C_1 and C_2 with shortest length, otherwise we simply replace the line A_1A_2 with a strictly shorter line.

Now, we build the minimum spanning tree over the network points. At each step, we look for the shortest connection between the network points. At some point, we must connect C_1 and C_2 with a line. Since the line A_1A_2 is a connection with shortest length between C_1 and C_2, we can use that line. □

Theorem 6 is helpful in building Steiner trees: we do not connect network points together in a Steiner tree unless they are connected in one of the network's minimum spanning trees.

Theorem 7 (The Steiner Ratio) *The length of a network's Steiner minimal tree cannot be less than $\sqrt{3}/2$ times the length of the network's minimum spanning tree.*

Theorem 7 was conjectured in 1968 in [12], but was not proved until 1990 in [6, 7] (as stated in [15]). The proof is quite complicated and involves defining a new class of trees, minimal hexagonal trees, whose points correspond to the so-called "minimal critical points" for any general network [15]. We omit the proof here as it adds nothing to our discussion. However, Theorem 7 gives us a lower bound for the length of a network's Steiner minimal tree and any Steiner tree that attains this lower bound is known to be optimal; the upper bound is given by the length of the corresponding minimum spanning tree [12].

CHAPTER 3

WELL-KNOWN STEINER MINIMAL TREES

We have noted that the general case of the Steiner Problem is difficult, if not impossible, to solve. However, we can rather easily solve "small" cases. This chapter presents the two best-known Steiner minimal trees — the Steiner minimal tree over a triangle and the Steiner minimal tree over a square.

3.1 The Steiner minimal tree over a triangle

Clearly, the smallest possible Steiner Problem involves building a Steiner minimal tree over three points. (The case for two points is trivial: simply connect the two points with a straight line.) The solution for the three-points case is well-known, and its construction offers insight into solving larger cases.

3.1.1 The basic triangle construction

(This construction is given in [4, 12].) Given points A, B, and C in the plane, we form $\triangle ABC$. Letting $\angle B$ denote the largest angle in the triangle gives us two cases:

1. $\angle B \geq 120°$.

2. All the angles in $\triangle ABC$ are less than $120°$.

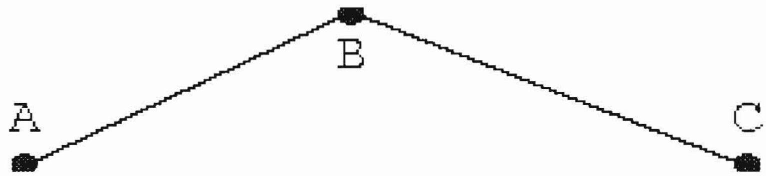


Figure 3.1 The Steiner minimal tree for $\angle B \geq 120°$.

Case 1 is trivial: the Steiner minimal tree over $\triangle ABC$ actually corresponds to the minimum spanning tree for the triangle, which is built from lines $\overline{AB}$ and $\overline{BC}$ (Figure 3.1). Since $\angle B \geq 120°$, Lemma 1 implies we cannot shorten the minimum spanning tree by adding in a Steiner point. (If we go through the construction given below, the constructed Steiner point will lie outside $\triangle ABC$ [4], thus violating Theorem 5 and resulting in a longer tree.)

For Case 2, we begin by constructing two 120° arcs, one with chord $\overset{\frown}{AB}$ and one with chord $\overset{\frown}{BC}$. (We could have used any two of the triangle's edges for these chords.) These two arcs intersect at two points, vertex B and an interior point of the triangle, which we have labeled point S (see Figure 3.2). Point S is the Steiner point for $\triangle ABC$, with the corresponding Steiner minimal tree consisting of line segments $\overline{AS}$, $\overline{BS}$, and $\overline{CS}$.

3.1.2 An alternative triangle construction

Although the construction in 3.1.1 is geometrically straight-forward, it can be complicated to implement and use effectively. For that reason, we describe here a slightly

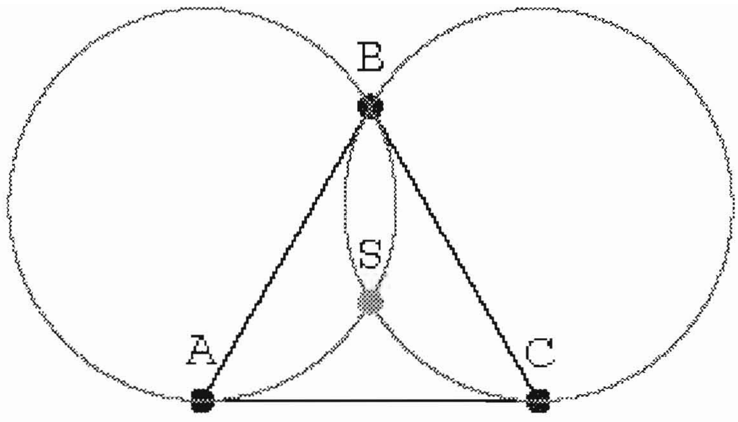


Figure 3.2 The two 120° arcs on $\triangle ABC$ intersect at one vertex of the triangle and at the Steiner point for the triangle.

different triangle construction (from [12]), that will enable us to more easily find a general expression for the length of any triangle Steiner tree.

First, find the point B' in the exterior $\triangle ABC$, where $\triangle ABB'$ is an equilateral triangle. Circumscribing $\triangle ABB'$ creates the 120° arc with chord $\overset{\frown}{AB}$. Then the Steiner point S is the intersection of arc $\overset{\frown}{AB}$ and the the line segment $\overline{B'C}$ (see Figure 3.3). Further, the length of the Steiner tree is given by the length of the line segment $\overline{B'C}$. (See [5] for proofs of these facts.)

3.2 The Steiner minimal tree over a square

Perhaps the second best-known Steiner minimal tree is that over a square, shown in Figure 3.4. For a unit square, this Steiner tree has length $1 + \sqrt{3}$.

The actual construction of the Steiner tree over a square is often left to empirical means. (We develop a geometric construction for the more general rhombus in Section

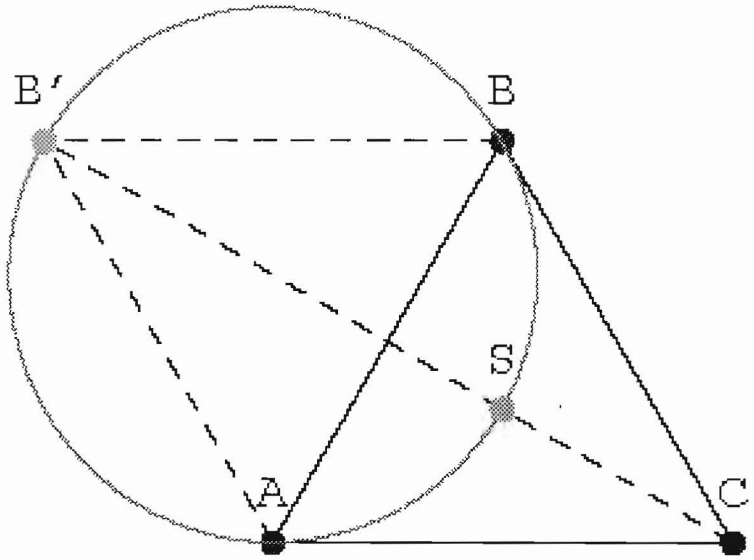


Figure 3.3 The Steiner point S is the intersection of arc $\widehat{AB}$ and line segment $\overline{B'C}$.

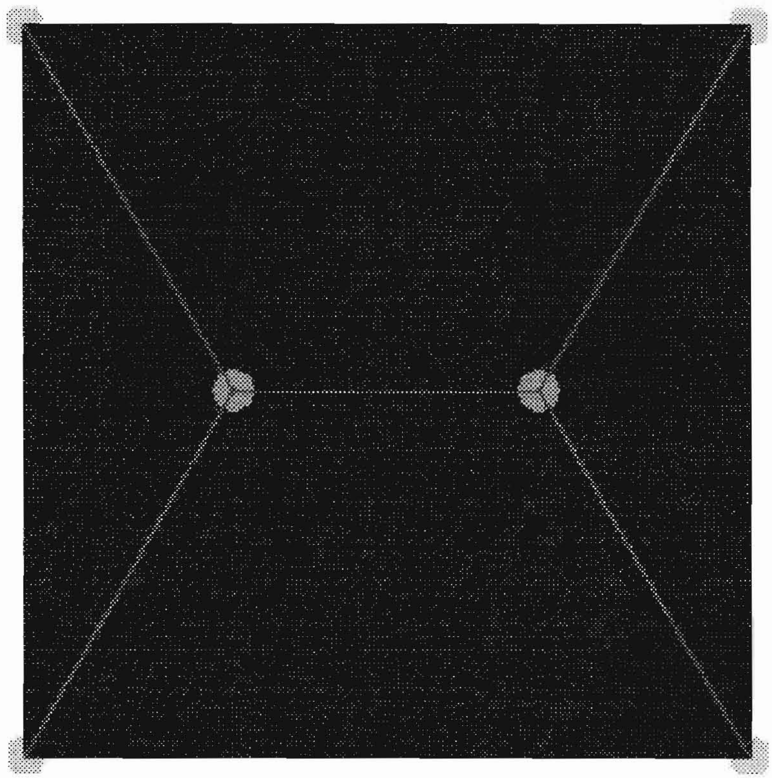


Figure 3.4 The Steiner minimal tree over a square.

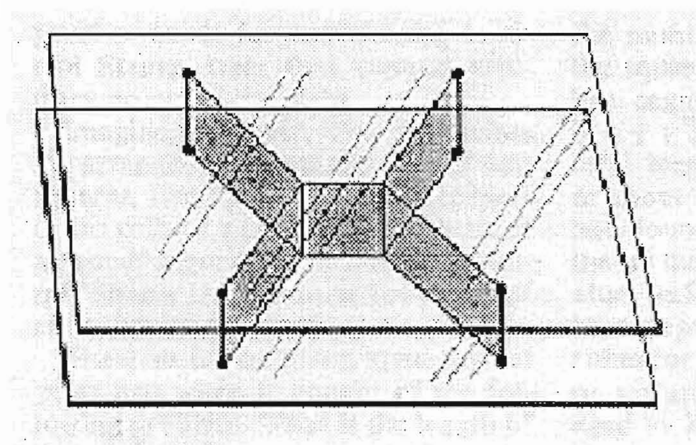


Figure 3.5 Finding the Steiner minimal tree empirically: the resulting soap film [9].

4.1.) Gardner [9] describes a method for finding the square's Steiner minimal tree by making use of two parallel sheets of Plexiglas joined by perpendicular rods that correspond to the network points. (This process is also described in [4, pp. 391–392].) The assembly is dipped into a soap solution, and when it is lifted out, the resulting soap film forms the Steiner minimal tree, as shown in Figure 3.5. This occurs because the film's surface will shrink to minimal area. However, this empirical process does not work with all networks: depending upon the number of points and the network configuration, the resulting soap film may not be stable [9]. Thus, although such a process may solve small cases, it will not solve the Steiner Problem in general.

CHAPTER 4

PRELIMINARIES FOR GENERALIZED CHECKERBOARDS: SINGLE STEINER TREES

Due to the large number of network points in a generalized checkerboard, building a full Steiner tree over such a network is difficult. Instead, we look to build a Steiner tree that is the union of full Steiner trees. Chung, Gardner and Graham followed this course in [2] when they looked at standard checkerboards (checkerboards made of squares) by mainly using copies of the Steiner tree over a unit square. In order to generalize their work, we construct the Steiner tree over a unit rhombus and find its length in terms of the small angle of the rhombus. Since Steiner trees over isosceles triangles are used as well (see Figure 1.1), we also find the length for these trees.

Unless otherwise noted, all figures are drawn using *Mathematica* programs which can be found in Appendix A.

4.1 The Steiner tree over a rhombus

In this section, we develop a geometric construction for a full Steiner tree over a rhombus. (The Steiner tree over any convex quadrilateral is known [17].) A typical full rhombus Steiner tree is shown in Figure 4.1.

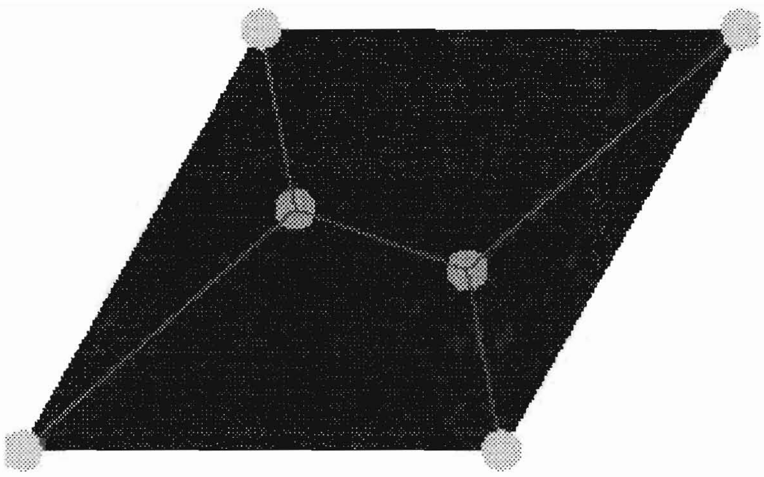


Figure 4.1 The Steiner tree over a rhombus.

From the figure, it appears the full Steiner tree passes through the center of the rhombus. Indeed, we prove this for any full rhombus Steiner tree.

Theorem 8 *A full Steiner tree over a rhombus passes through the center of the rhombus.*

Proof: We know from [17] that there are only two possible full Steiner trees over a rhombus (see Figure 4.2). Further due to symmetry, these two trees are actually isomorphic, so we concentrate on the first tree.

(Refer to Figure 4.3.) The triangles $\triangle AS_1B$ and $\triangle ES_2D$ must be congruent because

$$\angle AS_1B \cong \angle ES_2D,$$

$$\begin{aligned} \angle AS_1S_2 \cong \angle ES_2S_1 \quad &\Rightarrow \quad \overleftrightarrow{AS_1} \parallel \overleftrightarrow{ES_2} \\ &\Rightarrow \quad \angle BAS_1 \cong \angle DES_2, \end{aligned}$$

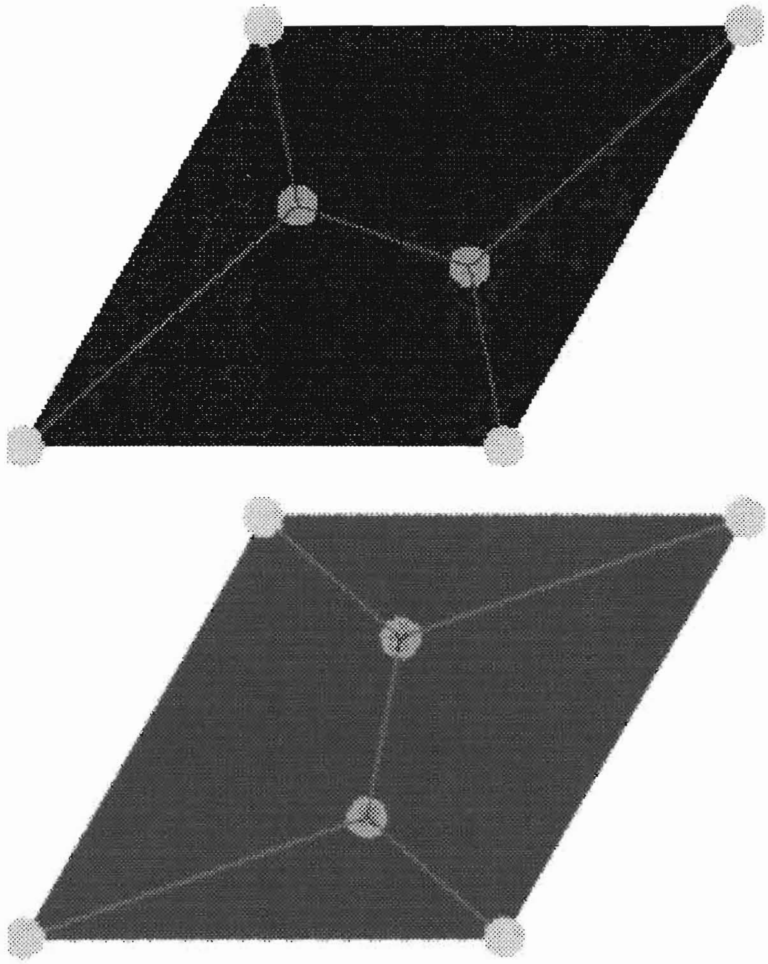


Figure 4.2 The two possible full Steiners tree over a rhombus.

$$\angle BS_1S_2 \cong \angle DS_2S_1 \;\Rightarrow\; \overleftrightarrow{BS_1} \parallel \overleftrightarrow{DS_2}$$

$$\Rightarrow\; \angle ABS_1 \cong \angle EDS_2,$$

and $|AB| = |ED| = 1$.

Now, $\triangle BS_1C \cong \triangle DS_2C$ since

$$\triangle AS_1B \cong \triangle ES_2D \;\Rightarrow\; |BS_1| = |DS_2|,$$

$$\overleftrightarrow{AB} \parallel \overleftrightarrow{ED} \;\Rightarrow\; \angle DBA \cong \angle BDE,$$

$$\angle DBA \cong \angle BDE \text{ and } \angle ABS_1 \cong \angle EDS_2 \;\Rightarrow\; \angle S_1BD \cong \angle S_2DB,$$

$$\angle S_1CB \cong \angle S_2CD$$

and

$$\angle BS_1S_2 \cong \angle DS_2S_1.$$

But this implies $|BC| = |DC|$. So, $\overline{S_1S_2}$ bisects the diagonal of the rhombus $\overline{BD}$ at point C, thus implying point C is the center of the rhombus. Therefore, a full Steiner tree over a rhombus passes through the center of the rhombus. □

To construct the full Steiner tree over the rhombus, divide the rhombus into four triangles by drawing in the diagonals of the rhombus and including the center point.

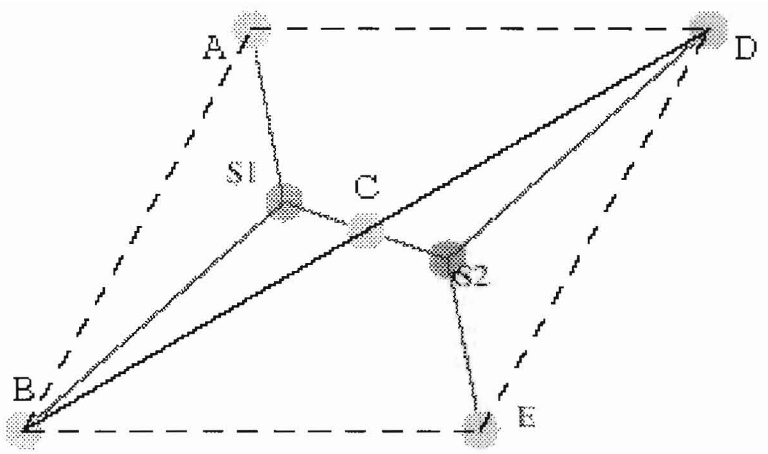


Figure 4.3 The Steiner tree over a rhombus passes through the center of the rhombus.

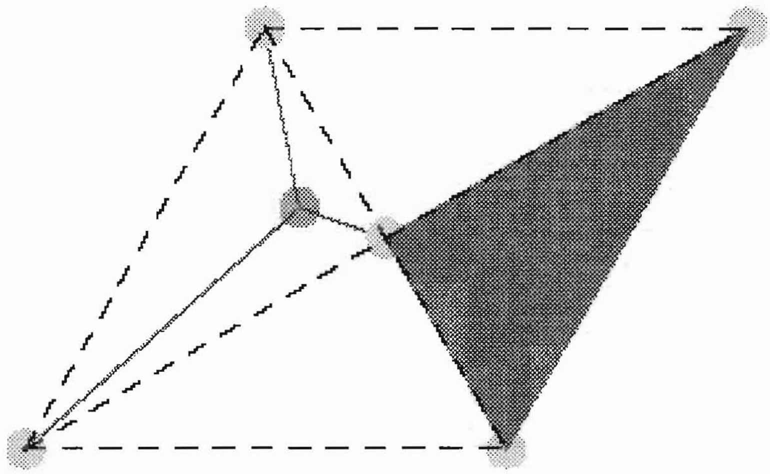


Figure 4.4 Half the rhombus Steiner tree.

Because the full Steiner tree passes through the center, we can pick one of these four triangles, and build its corresponding triangle Steiner tree as described in Section 3.1.2. Since all four triangles are congruent by basic properties of the rhombus, we can choose any of the four. This results a Steiner tree covering half the rhombus, as shown in Figure 4.4.

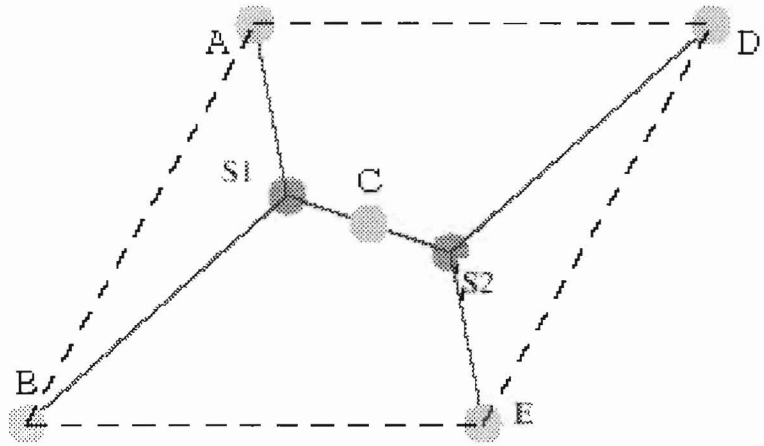


Figure 4.5 The two "half" Steiner trees meet to form a straight line.

To finish the Steiner tree for a rhombus, we repeat this process on the "opposite" triangle (the shaded triangle in Figure 4.4). This gives the Steiner tree over the entire rhombus (Figure 4.5). Because $\overleftrightarrow{AS_1} \parallel \overleftrightarrow{ES_2}$ and $\angle AS_1C$ and $\angle ES_2C$ are both 120°, $\overleftrightarrow{S_1C} \parallel \overleftrightarrow{S_2C}$. Further, $\overleftrightarrow{S_1C}$ and $\overleftrightarrow{S_2C}$ both pass through point C. So $\overleftrightarrow{S_1C}$ and $\overleftrightarrow{S_2C}$ lie on the same line, and the two "half" Steiner trees do meet to form a straight line at point C.

To find a general expression for length of the Steiner tree, we impose coordinate axes on the rhombus as follows: put the left side of the rhombus on y-axis with the lower left point of the rhombus at point (0, 0) (see Figure 4.6). Then, the base of the rhombus forms some angle with the x-axis. The peak (point B') of the equilateral triangle formed outside of the rhombus during the construction does not depend on α, the small angle of the rhombus, while the center of the rhombus (point C) does depend on α. Using

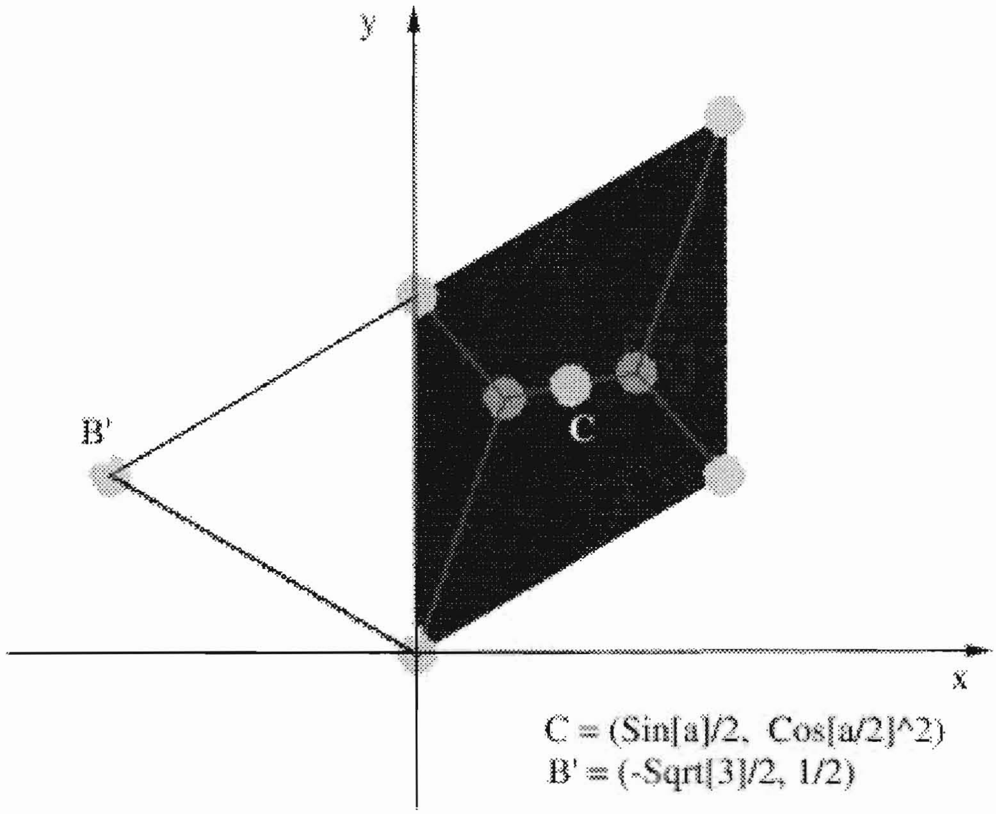


Figure 4.6 Coordinatizing the rhombus Steiner tree.

standard geometry, the coordinates of B' and C are

$$B' = (-\sqrt{3}/2, 1/2)$$

$$C = (\frac{\sin\alpha}{2}, \cos^2(\alpha/2)).$$

The distance between B' and C gives the length of "half" the Steiner tree; geometry and algebra can be used to show that this has the length $\frac{\sqrt{4+2\sqrt{3}\sin\alpha}}{2}$. Thus, the length of the corresponding rhombus Steiner tree is given by

$$r(\alpha) = \sqrt{4 + 2\sqrt{3}\sin\alpha} \tag{4.1}$$

where α is the small angle of the rhombus in degrees.

4.2 The Steiner tree over an isosceles triangle

We can build the Steiner tree over any isosceles triangle using the construction of Section 3.1.2. However, this construction does not give us the length of the resulting Steiner tree without calculating the lengths of each of the lines in the tree. We need a general expression (along the lines of Equation 4.1) for the length of the Steiner tree over an isosceles triangle in terms of the angle between the two equal sides.

Figure 4.7 shows the Steiner tree, given by edges AS, BS, and CS, over a typical isosceles triangle. Without loss of generality, we may assume that $|AB| = |BC| = 1$ and that $\angle B$ has measure α. (Note that $\angle B$ corresponds directly to the small angle of the

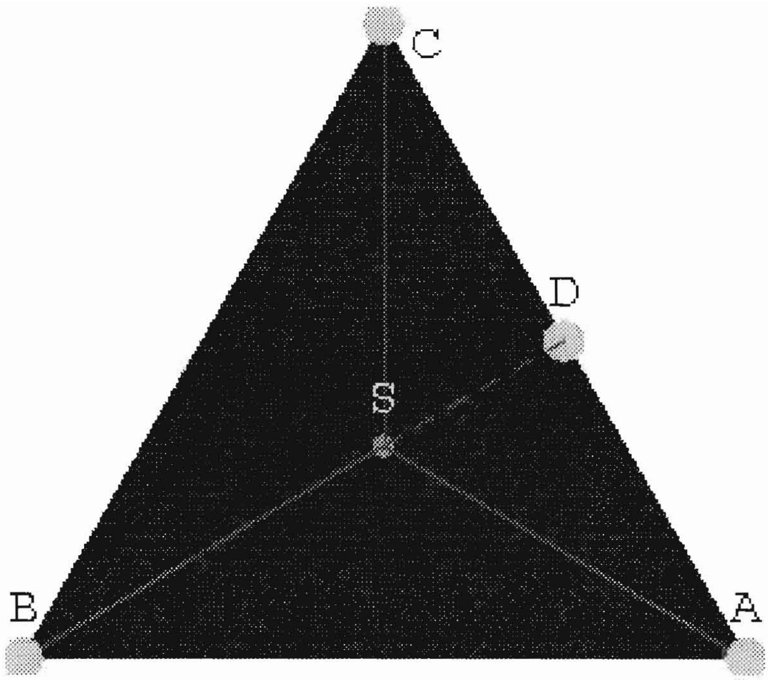


Figure 4.7 The Steiner tree over an isosceles triangle.

rhombus.) Because $|AB| = |BC|$, we expect points A and C to pull on B with equal force. Therefore, the line segment BS bisects $\angle B$. Further, the extended line segment BD bisects $\angle ASC$, implying that both $\angle ASD$ and $\angle CSD$ are $60°$ angles.

Thus $\triangle CSD \cong \triangle ASD$ and simple geometry implies that $|BD| = \cos(\alpha/2)$, $|AD| = |CD| = \sin(\alpha/2)$, $|AS| = |CS| = \frac{\sin(\alpha/2)}{\sin 60°}$, and $|SD| = \sin(\alpha/2)\tan 30°$. So, the length of the Steiner tree is given by

$$\begin{aligned}
t(\alpha) &= |BS| + |AS| + |CS| \\
&= |BD| - |SD| + 2|CS| \\
&= \cos(\alpha/2) - \sin(\alpha/2)\tan 30° + 2\frac{\sin(\alpha/2)}{\sin 60°}
\end{aligned}$$

$$= \cos(\alpha/2) + \sqrt{3}\sin(\alpha/2)$$

Note that $t(\alpha)$ is now expressed as a linear combination of sine and cosine functions. We can further simplify $t(\alpha)$ using properties of sines and cosines to:

$$t(\alpha) = 2\cos(\alpha/2 - 60°) \tag{4.2}$$

4.3 Minimality of rhombus and triangle Steiner trees

We know that the triangle construction in Section 3.1.2 returns the Steiner minimal tree [12], but what about the rhombus construction?

The Steiner minimal tree over a rhombus will consist of zero, one, or two Steiner points. Consider each case individually.

- Zero Steiner points: A Steiner tree with no Steiner points corresponds to the minimum spanning tree for the network. Let α be the measure of the small angle of the rhombus. When $\alpha \geq 60°$, the minimum spanning tree consists of three edges of the rhombus (see Figure 4.8) and has length 3. However, there exists at least one angle (corresponding to the small angle of the rhombus) whose measure is less than 120°. Lemma 1 then implies that this cannot be the Steiner minimal tree.

- One Steiner point: We have two possible cases, shown in Figure 4.9, both with length $t(\alpha) + 1$. In both cases, the angle between the triangle Steiner tree and the

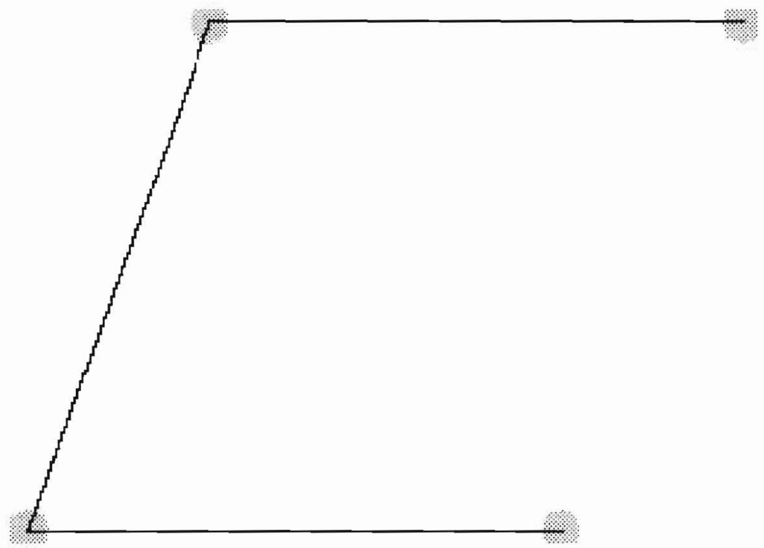


Figure 4.8 The spanning tree over a rhombus with $\alpha \geq 60°$.

rhombus edge is less than 120°, implying (by Lemma 1) that neither tree can be the Steiner minimal tree.

- Two Steiner points: We build this Steiner tree using the rhombus construction given above. All angles in this Steiner tree are equal to 120°. Further, since all other possible trees have been eliminated, this is the Steiner minimal tree over the rhombus for $\alpha \geq 60°$.

Therefore, the Steiner minimal tree over a rhombus with $\alpha \geq 60°$ is given by the rhombus construction. We now show that this is the case for all rhombuses:

Theorem 9 *The rhombus construction returns the Steiner minimal tree over a rhombus.*

Proof: Introductory remarks imply we only need to consider the case for $\alpha < 60°$. First consider the tree with zero Steiner points, i.e. the minimum spanning tree (given in Figure 4.10). One edge of the minimum spanning tree corresponds to the short diagonal

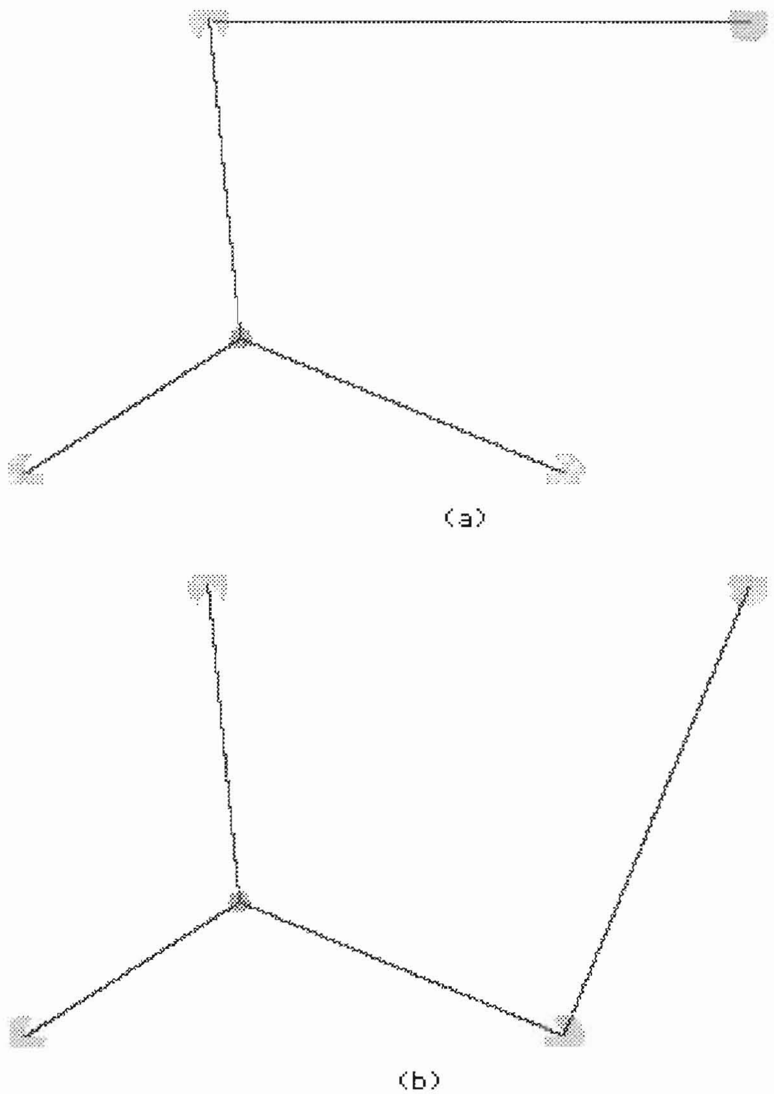


Figure 4.9 The rhombus Steiner trees with only one Steiner point for $\alpha \geq 60°$.

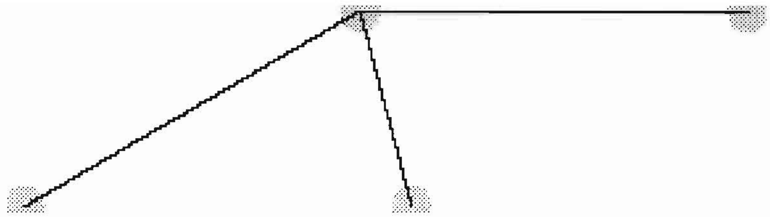


Figure 4.10 The spanning tree over a rhombus with $\alpha < 60°$.

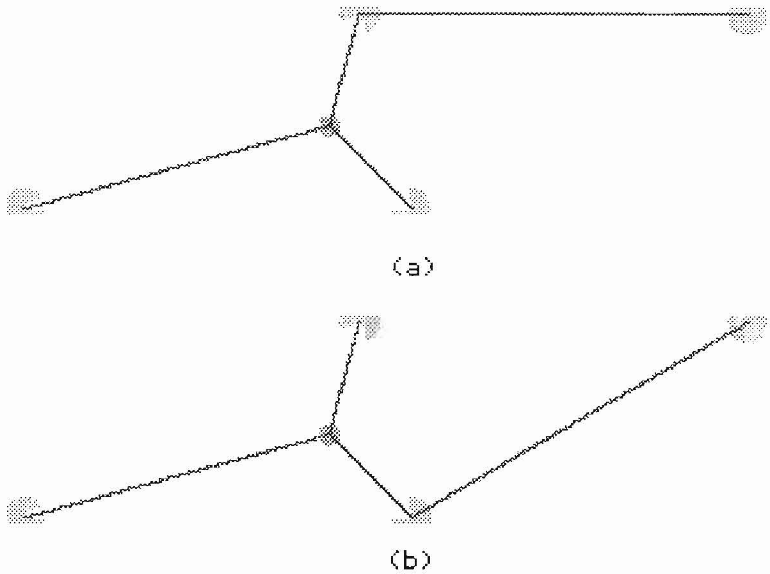


Figure 4.11 The rhombus Steiner trees with only one Steiner point for $\alpha < 60°$.

of the rhombus. But, this diagonal bisects the large angle of the rhombus, and thus the angles in the minimum spanning tree are less than 120°. By Lemma 1, the minimum spanning tree cannot be the Steiner minimal tree for the rhombus.

Now, consider the tree containing one Steiner point. We have two possible cases, shown in Figure 4.11, both with length $t(\alpha) + 1$. We compare both trees to the full rhombus Steiner tree. In order to see which is bigger, define the function $f(\alpha)$ to be the difference between the length of the trees with one Steiner point and the length of the

full Steiner tree:

$$\begin{aligned} f(\alpha) &= t(\alpha) + 1 - r(\alpha) \\ &= 1 + 2\cos(60^\circ - \alpha/2) - \sqrt{4 + 2\sqrt{3}\sin\alpha} \end{aligned}$$

(see the plot given in Figure 4.12). Note that $f(\alpha)$ is a continuous function and that $f(0) = 0$. In addition, $f'(\alpha) = \sin(60^\circ - \alpha/2) - \sqrt{3}\cos\alpha/\sqrt{4 + 2\sqrt{3}\sin\alpha}$. Using trigonometric properties and the fact that $0^\circ < \alpha < 60^\circ$, it can be shown that $f'(\alpha) > 0$ (see Figure 4.12 and Appendix B). So, by standard calculus, $f(\alpha)$ is strictly increasing when $0 < \alpha < 60^\circ$. This implies that $f(\alpha) > 0$ for $0 < \alpha < 60^\circ$, further implying the length of either triangle tree is longer than the length of full Steiner tree. Therefore, the full rhombus Steiner tree is a Steiner minimal tree for the rhombus. □

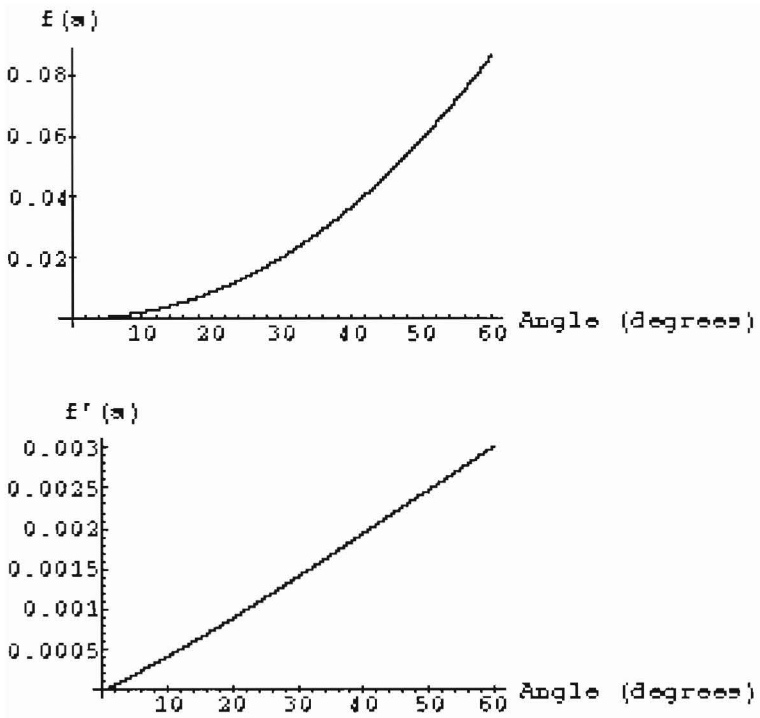


Figure 4.12 The plots of $f(\alpha)$ and $f'(\alpha)$.

CHAPTER 5

GENERALIZED "POWERS OF 2" CHECKERBOARDS

Now that we have found the Steiner minimal trees that act as the building blocks for our generalized checkerboards, we move onto filling the checkerboards. We begin by looking at the "nicest" checkerboards: the "powers of 2" checkerboards, so named because they are lattices of $2^k \times 2^k$ points or, as we refer to them, $(2^k - 1) \times (2^k - 1)$ layers of rhombuses (for $k \geq 1$). We consider the two most obvious ways of filling such a generalized checkerboard: filling it with Steiner trees over a rhombus and filling it with Steiner trees over an isosceles triangle.

Throughout the rest of this paper, ℓ will denote the number of layers of rhombuses in the checkerboard and α will denote the measure of the small angle of the rhombus.

5.1 The rhombus scheme

First, we describe the technique for filling "powers of 2" checkerboards with rhombus Steiner trees. Then we count the number of rhombus Steiner trees needed to fill any "powers of 2" grid. Finally, we use this information to find the length of the tree.

5.1.1 Filling a grid of rhombuses

To fill the "powers of 2" generalized checkerboard with rhombuses, we use the technique in [2], given for filling "powers of 2" grids of squares. The technique is recursive: the tree for the grid of $\ell = 2^k - 1$ layers is based on the tree for the grid of $\ell = 2^{k-1} - 1$ layers.

To fill a generalized checkerboard of $\ell = 2^k - 1$ layers with Steiner trees over a rhombus:

1. For $k = 1$, we have a single rhombus. So, use the full Steiner tree over a rhombus.

2. For $k > 1$, begin with the tree on $2^{k-1} - 1$ layers.

3. "Spread" this tree over the larger grid of $2^k - 1$ layers.

4. Fill in with rhombuses.

An example of this technique is given in Figure 5.1.

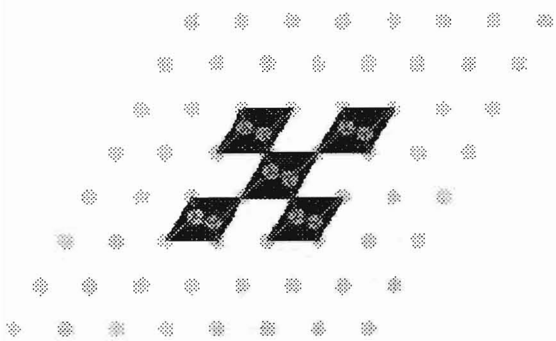
This technique has been proven to produce an optimal network when the "powers of 2" grid is built from squares [1].

5.1.2 Counting the number of rhombus Steiner trees needed

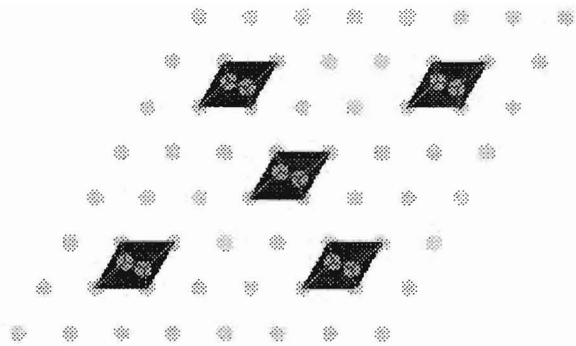
Theorem 10 *The number of rhombus Steiner trees needed to fill a grid of* $(2^k - 1) \times (2^k - 1)$ *layers of rhombuses is*

$$\sum_{j=1}^{k} 2^{2(j-1)} \tag{5.1}$$

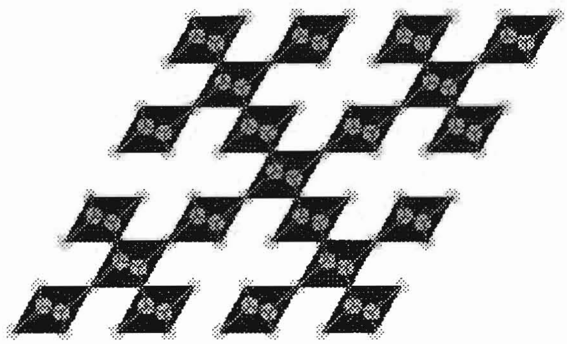
Proof: We use induction on k.



2. Tree for k = 2



3. Tree for k = 2 Spread Out



4. Final Rhombus Grid

Figure 5.1 Filling a grid of rhombuses: $k = 3$.

- Base Case: $k = 1$

  Setting $k = 1$ results in $\ell = 1$, a single rhombus, which we fill with one rhombus Steiner tree. Thus, we have the number of rhombus Steiner trees needed in Formula 5.1 since $2^0 = 1$.

- Induction Hypothesis: $k = n$

  Assume that a grid with $\ell = 2^n - 1$ layers requires

$$\sum_{j=1}^{n} 2^{2(j-1)}$$

  rhombus Steiner trees to fill.

- $k = n + 1$ Case

  The grid with $\ell = 2^{n+1} - 1$ layers contains a "stretched-out" version of the grid with $\ell = 2^n - 1$. So, we only need to count the number of "fill-in" rhombus Steiner trees needed. Consider a row containing "fill-in" rhombus Steiner trees. This row contains $(2^{n+1} - 1)$ cells, where every-other cell is filled with a rhombus Steiner tree (including both ends). So, the number of "fill-in" rhombus Steiner trees for this row is

$$\lceil \frac{2^{n+1} - 1}{2} \rceil = \frac{2^{n+1}}{2} = 2^n$$

By the same counting argument, we have 2^n such rows. Thus, the total number of "fill-in" rhombus Steiner trees needed for the grid with $\ell = 2^{n+1} - 1$ layers is

$$(2^n)(2^n) = 2^{2n} = 2^{2[(n+1)-1]}$$

So, the total number of rhombus Steiner trees need to fill a grid with $\ell = 2^{n+1} - 1$ layers is

$$\Big(\sum_{j=1}^{n} 2^{2(j-1)}\Big) + 2^{2[(n+1)-1]} = \sum_{j=1}^{n+1} 2^{2(j-1)}$$

as expected. $\square$

Thus, the number of rhombus Steiner trees needed to fill a $(2^k - 1) \times (2^k - 1)$ grid is

$$\sum_{j=1}^{k} 2^{2(j-1)} = \sum_{j=1}^{k} 4^{(j-1)}$$

However, this is the sum of a geometric series and so can be written in closed form using standard methods as:

$$\sum_{j=1}^{n+1} 4^{j-1} = \frac{4^k - 1}{3}. \tag{5.2}$$

The total length of the Steiner tree over a "powers of 2" generalized checkerboard using the rhombus scheme is then given by $\frac{4^k-1}{3} r(\alpha)$. But, since $\ell = 2^k - 1$, we have $k = \frac{\ln(\ell+1)}{\ln 2}$. So the total length of the Steiner tree can be written as a function of α and ℓ:

$$R(\alpha, \ell) = \frac{4^{\frac{\ln(\ell+1)}{\ln 2}} - 1}{3} r(\alpha). \tag{5.3}$$

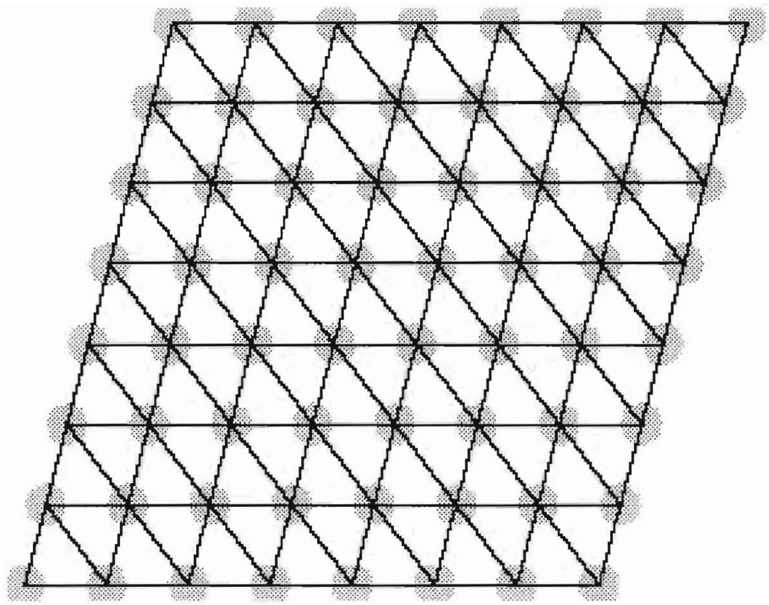


Figure 5.2 Looking at the generalized checkerboard as a grid of isosceles triangles.

5.2 The triangle scheme

Now that we have a method for filling the "powers of 2" grid with rhombuses, we would like to determine if this method returns the optimal network. One way to explore this question is to consider other ways of looking at and filling the grid (as we did earlier to prove the optimality of the rhombus Steiner tree and as Brazil et al. use in [1] to prove optimality for the square "powers of 2" grids). Another obvious way of looking at a generalized checkerboard is to consider it as a grid of isosceles triangles by adding in the diagonals of the rhombuses (Figure 5.2). How, then, do we fill the "triangle" grid with Steiner trees over isosceles triangles?

5.2.1 Filling a grid of isosceles triangles

The scheme presented is actually the general case for filling any grid with an *odd* number of layers. Like the rhombus scheme given in Section 5.1.1, this triangle scheme is also recursive.

To fill a generalized checkerboard of $(2c+1) \times (2c+1)$ layers of rhombus (for $c > 0$) with Steiner trees over an isosceles triangle:

1. Begin with a core of the Steiner tree over a single rhombus in the upper-left-hand corner. (This is needed to connect all of the network points.)
2. "Skip" a layer of rhombuses and wrap a layer of triangle Steiner trees around this "skipped" layer. (In the "skipped" layer, we do need to include one triangle Steiner tree in order to connect the core with the second layer.)
3. Using this tree as the new core, repeat step 2, wrapping in reverse order.
4. Repeat steps 2 and 3 until grid is filled.

An example of this technique is given in Figure 5.3.

5.2.2 Counting the number of triangle Steiner trees needed

Theorem 11 *The number of triangle Steiner trees needed to fill a grid of* $(2c+1) \times (2c+1)$ *layers of rhombuses is*

$$2\sum_{j=1}^{c}(2j+1) \tag{5.4}$$

(plus one rhombus Steiner tree for the core).

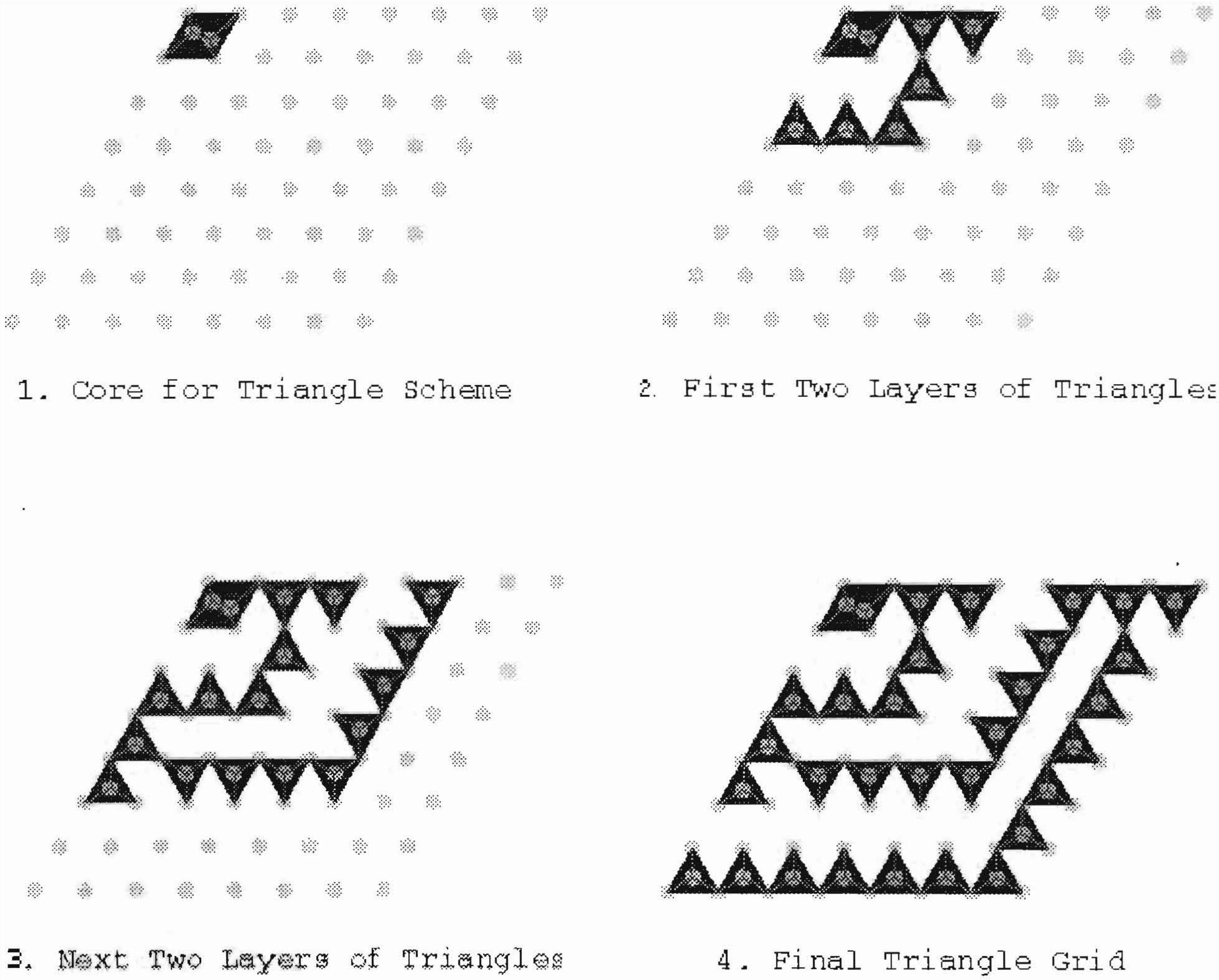


Figure 5.3 Filling a grid of isosceles triangles: $c = 3$.

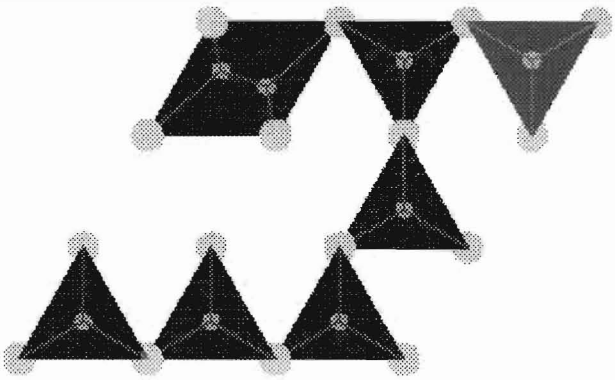


Figure 5.4 Triangle scheme: $c = 1$.

Proof: We use induction on c.

- Base Case: $c = 1$

  Setting $c = 1$ results in $\ell = 3$. We can fill this checkerboard with 6 triangle Steiner trees and 1 rhombus Steiner tree as shown in Figure 5.4. This gives the number of triangles needed for Formula 5.4 since $2(2 + 1) = 6$.

- Induction Hypothesis: $c = n$

  Assume that a grid with $\ell = 2n + 1$ layers requires

$$2\sum_{j=1}^{n}(2j+1)$$

  triangle Steiner trees (and one rhombus Steiner tree) to fill.

- $c = n + 1$ Case

  We start with the grid for $\ell = 2n + 1$ layers and add two layers to it to make the grid for $\ell = 2(n + 1) + 1$. Now, we count the number of triangle Steiner trees

added. In the second layer, we add one triangle Steiner tree to each rhombus cell. The second layer consists of $2[2(n+1)+1]-1$ rhombus cells. (We subtract 1 because the lower-right-hand corner counts in both the row and column.) So we add $2[2(n+1)+1]-1$ triangle Steiner trees in the second layer. However, in order to connect this second layer with the current tree, we need to add one triangle Steiner tree in the first layer. Thus, the grid for $\ell = 2(n+1)+1$ layers requires

$$\left(2\sum_{j=1}^{n}(2j+1)\right)+1+2[2(n+1)+1]-1 = 2\sum_{j=1}^{n+1}(2j+1)$$

triangle Steiner trees (plus one rhombus Steiner tree) to fill, as we expected. □

Using the fact that $\sum_{j=1}^{c}(2j-1) = c^2$, a grid with $\ell = 2c+1$ layers requires

$$2\sum_{j=1}^{c}(2j+1) = 2[(c+1)^2-1] \tag{5.5}$$

triangle Steiner trees and one rhombus Steiner tree to fill.

The total length of the Steiner tree over a "powers of 2" generalized checkerboard using the triangle scheme is then given by $2[(c+1)^2-1]t(\alpha)+r(\alpha)$. But, since $\ell = 2c+1$, we have $c = \frac{\ell-1}{2}$. So the total length of the Steiner tree can be written as a function of α and ℓ:

$$T(\alpha,\ell) = 2[(\frac{\ell-1}{2}+1)^2-1]t(\alpha)+r(\alpha). \tag{5.6}$$

5.3 Comparing the rhombus scheme to the triangle scheme

We now have two different schemes for building Steiner trees on "powers of 2" girds. Which scheme is better? To explore the answer to this question, we look at the 3×3 checkerboard. For $\alpha = 90°$ the rhombus scheme has length $5r(90°) = 13.66025\ldots$ while the triangle scheme has (longer) length $6t(90°) + r(90°) = 14.32316\ldots$ (see Figure 5.5). On the other hand, for $\alpha = 60°$ the rhombus scheme has length $5r(60°) = 13.22875\ldots$ while triangle scheme has (shorter) length $6t(60°) + r(60°) = 13.03805\ldots$ (see Figure 5.6).

To answer the question more fully, we look for the *crossover angle* — the angle at which the better of the two schemes changes — by plotting the difference between the rhombus scheme (Equation 5.3) and the triangle scheme (Equation 5.6) for various sized "powers of 2" checkerboards with *Mathematica* (the code and its output are given in Appendix B). The crossover angle is found when the difference is zero:

$$R(\alpha, \ell) - T(\alpha, \ell) = 0$$

$$\Downarrow$$

$$\frac{4^{\frac{\ln(\ell+1)}{\ln 2}} - 1}{3} r(\alpha) - \left(2[(\frac{\ell - 1}{2} + 1)^2 - 1]t(\alpha) + r(\alpha)\right) = 0 \qquad (5.7)$$

The result appears to show the same crossover angle for every "powers of 2 " checkerboard (see Figure 5.7 and Appendix B).

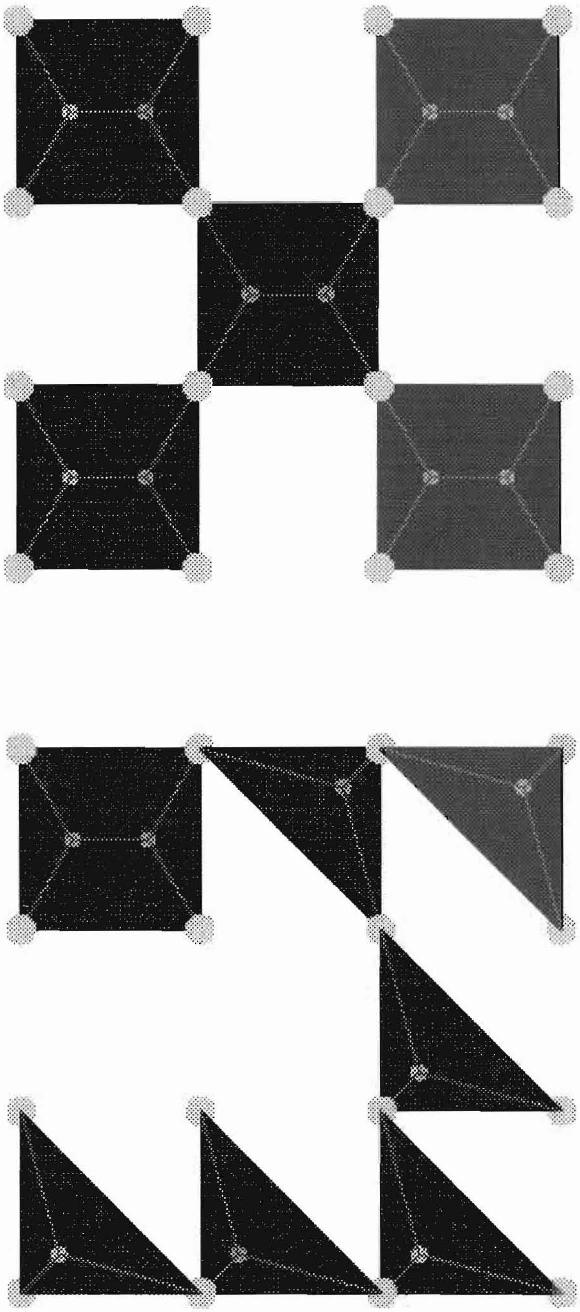


Figure 5.5 The 3×3 checkerboard with $\alpha = 90°$.

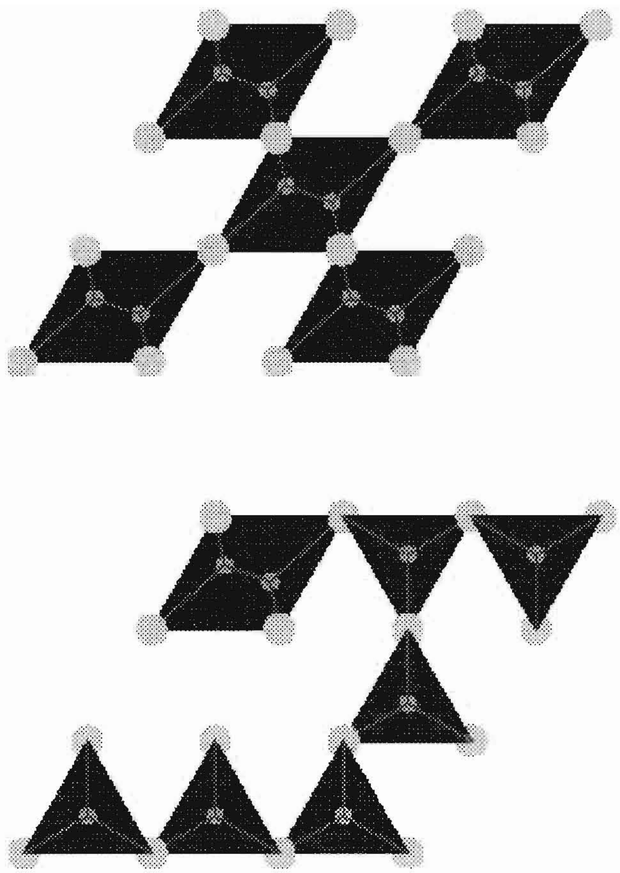


Figure 5.6 The 3×3 checkerboard with $\alpha = 60°$.

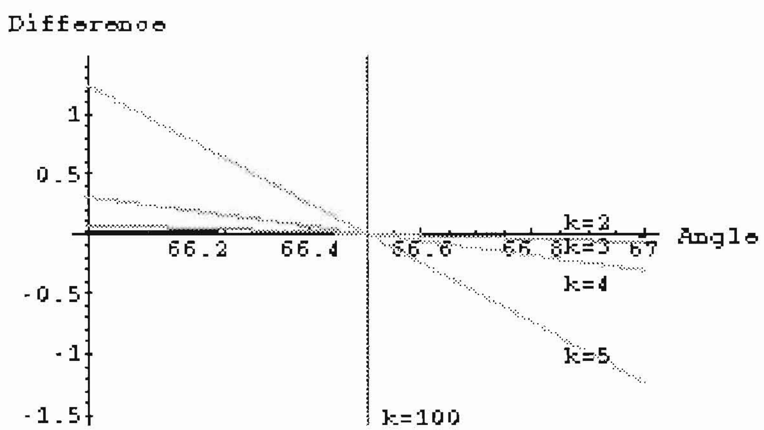


Figure 5.7 Each "powers of 2" checkerboard appears to have the same crossover angle.

To verify what appears graphically, we solve Equation 5.7 for α:

$$\begin{aligned}
\frac{4^{\frac{\ln(\ell+1)}{\ln 2}}-1}{3} r(\alpha)-\left(2[(\frac{\ell-1}{2}+1)^{2}-1] t(\alpha)+r(\alpha)\right) &= 0 \\
&\Downarrow \\
\frac{4^{\frac{\ln[\ell+1]}{\ln 2}}-1}{3} r(\alpha)-(2[(\frac{\ell-1}{2}+1)^{2}-1] t(\alpha)+r(\alpha)) &= 0 \\
&\Downarrow \\
\frac{[\ell+1]^{2}-1}{3} r(\alpha)-(\frac{[\ell+1]^{2}-4}{2} t(\alpha)+r(\alpha)) &= 0 \\
&\Downarrow \\
\frac{[\ell+1]^{2}-4}{3} r(\alpha)-\frac{[\ell+1]^{2}-4}{2} t(\alpha) &= 0 \\
&\Downarrow \\
([\ell+1]^{2}-4)(\frac{r(\alpha)}{3}-\frac{t(\alpha)}{2}) &= 0. \qquad (5.8)
\end{aligned}$$

Setting $[\ell+1]^2-4=0$ gives $\ell=-3$ or $\ell=1$, the special case in which the grid consists of one single rhombus. Thus, Equation 5.8 implies that the crossover angle does *not* depend upon the size of the grid as long as $\ell>1$. Setting $\frac{r(\alpha)}{3}-\frac{t(\alpha)}{2}=0$ gives the following:

$$\begin{aligned}
\frac{r(\alpha)}{3}-\frac{t(\alpha)}{2} &= 0 \\
&\Downarrow \\
\frac{\sqrt{4+2\sqrt{3}\sin\alpha}}{3}-\frac{2\cos(\alpha/2-60^{\circ})}{2} &= 0 \\
&\Downarrow \\
4+2\sqrt{3}\sin\alpha &= 9\cos^{2}(\alpha/2-60^{\circ})
\end{aligned}$$

$$\Downarrow$$

$$4 + 2\sqrt{3}\sin\alpha = (9/2)(1 + \cos(\alpha - 120°)$$

$$\Downarrow$$

$$4 + 2\sqrt{3}\sin\alpha = (9/2)(1 + \cos\alpha\cos 120° + \sin\alpha\sin 120°)$$

$$\Downarrow$$

$$(9/4)\cos\alpha = (1/2) + (\sqrt{3}/4)\sin\alpha \tag{5.9}$$

Because α is the small angle of the rhombus, we know $0 \leq \alpha \leq 90°$. So, $0 \leq \sin\alpha \leq 1$ and $0 \leq \cos\alpha \leq 1$. Setting $x = \sin\alpha$ lets us rewrite Equation 5.9 as

$$(9/4)\sqrt{1 - x^2} = (1/2) + (\sqrt{3}/4)x. \tag{5.10}$$

since $\cos\alpha = \sqrt{1 - x^2}$. Solving Equation 5.10 gives

$$x = \frac{-4\sqrt{3} - 72\sqrt{5}}{168} < 0$$

or

$$x = \frac{-4\sqrt{3} + 72\sqrt{5}}{168} > 0.$$

Because $x = \sin\alpha > 0$, we choose the positive root, and thus

$$\sin\alpha = x = \frac{-4\sqrt{3} + 72\sqrt{5}}{168}.$$

So, the crossover angle for any "powers of 2" checkerboards is

$$\arcsin[\frac{-4\sqrt{3}+72\sqrt{5}}{168}] \text{ radians} \approx 66.502222702\ldots^\circ.$$

We have now proven the following theorem:

Theorem 12 *For "powers of 2" generalized checkerboards with $\ell > 1$ and $\alpha < 66.5022\ldots^\circ$, the triangle scheme is better; for "powers of 2" generalized checkerboards with $\ell > 1$ and $\alpha > 66.5022\ldots^\circ$, the rhombus scheme is better.*

CHAPTER 6

OTHER GENERALIZED CHECKERBOARDS

6.1 The rhombus schemes

In [2] Chung, Gardner and Graham conjectured formulas for the lengths of Steiner trees over other square grids. These formulas involve more elaborate wrapping schemes around various cores (similar to the triangle scheme given in Section 5.2.1). We directly generalize their techniques to grids of rhombuses, resulting the formulas given in Table 6.1.

For the row marked with a ‡, the original formula given in [2] is incorrect; the correct formula was obtained from [13].

Grid Size (in ℓ)	Length of Steiner Tree	$R(\alpha, \ell)$
$6k$	$(12k^2 + 4k - 1)r(\alpha) + 3$	$(12(\frac{\ell}{6})^2 + 4\frac{\ell}{6} - 1)r(\alpha) + 3$
† $6k + 1$	$(12k^2 + 8k - 2)r(\alpha) + l(\alpha)$	$(12(\frac{\ell-1}{6})^2 + 8\frac{\ell-1}{6} - 2)r(\alpha) + l(\alpha)$
$6k + 2$	$(12k^2 + 12k + 2)r(\alpha) + 2$	$(12(\frac{\ell-2}{6})^2 + 12\frac{\ell-2}{6} + 2)r(\alpha) + 2$
† $6k + 3$	$(12k^2 + 16k + 2)r(\alpha) + l(\alpha)$	$(12(\frac{\ell-3}{6})^2 + 16\frac{\ell-3}{6} + 2)r(\alpha) + l(\alpha)$
$6k + 4$	$(12k^2 + 20k + 7)r(\alpha) + 3$	$(12(\frac{\ell-4}{6})^2 + 20\frac{\ell-4}{6} + 7)r(\alpha) + 3$
‡ $6k + 5$	$(12(k + 1)^2 - 1)r(\alpha) + t(\alpha)$	$(12(\frac{\ell-5}{6} + 1)^2 - 1)r(\alpha) + t(\alpha)$

Table 6.1 The rhombus schemes for other generalized checkerboards.

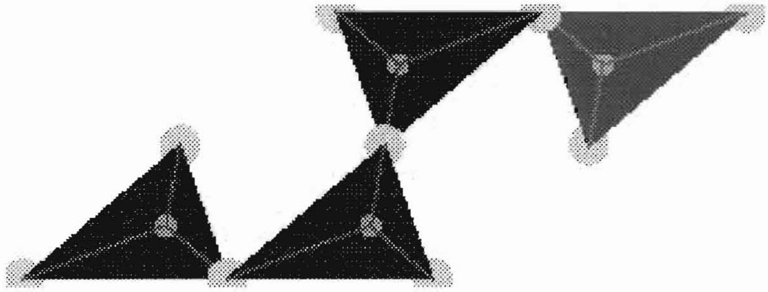


Figure 6.1 The core used for filling an even-layered grid with isosceles triangle Steiner trees.

The rows in Table 6.1 marked with a † denote lengths that make use of the term $l(\alpha)$, where $l(\alpha)$ represents the length of a Steiner tree over a row of four rhombuses (referred to as a 1×4 ladder). An explicit formula is known for such a row of squares [2, 3], but we do not have a direct generalization for the general rhombus case. See Section 7.1 for further details.

6.2 The triangle schemes

The general triangle scheme for any grid with an odd number of layers was developed in Section 5.2.1. The scheme for filling any grid with an even number of layers is similar. But rather than using a core of a single rhombus Steiner tree, we use the 2×2 core filled with triangles shown in Figure 6.1. This results in the following theorem.

Theorem 13 *The number of triangle Steiner trees needed to fill a grid of $2c \times 2c$ layers of rhombuses is*

$$4 \sum_{j=1}^{c} j. \tag{6.1}$$

The proof for Theorem 13 uses induction on c and is similar to that of Theorem 11. Using the fact that $\sum_{j=1}^{c} j = (1/2)j(j+1)$, we can rewrite Equation 6.1 in closed form as follows:

$$4\sum_{j=1}^{c} j = 2c(c+1). \tag{6.2}$$

So, the total length of a Steiner tree over an even-layered grid using the triangle scheme is given by $2c(c+1)t(\alpha)$. But, since $\ell = 2c$, we have $c = \ell/2$. So, the length of the Steiner tree can be rewritten as

$$T(\alpha, \ell) = \ell(\frac{\ell}{2} + 1)t(\alpha). \tag{6.3}$$

6.3 Comparing the rhombus schemes to the triangle schemes

We now compare the rhombus schemes of Table 6.1 to the triangle schemes developed in Sections 5.2.1 and 6.2. For each grid size, we look for a crossover angle by examining the difference between the length of the Steiner tree formed by the corresponding rhombus scheme and the length of the Steiner tree formed by the corresponding triangle scheme. It turns out that whether ℓ is even or odd is important.

We begin by considering ℓ even.

Theorem 14 *For a generalized checkerboard with an even number of layers, the crossover angle is not constant but the approaches* $66.5022\ldots°$ *(the crossover angle achieved by the "powers of 2" grids) as* ℓ *approaches infinity.*

Proof: Consider each even-layered checkerboard:

- For $\ell = 6k$, we have:

$$
\begin{aligned}
(12(\frac{\ell}{6})^2 + 4\frac{\ell}{6} - 1)r(\alpha) + 3 &- \ell(\frac{\ell}{2} + 1)t(\alpha) = 0 \\
&\Downarrow \\
(2\ell^2 + 4\ell - 6)r(\alpha) + 18 &- (3\ell^2 + 6\ell)t(\alpha) = 0 \\
&\Downarrow \\
(2\ell^2 + 4\ell - 6 + \frac{18}{r(\alpha)})r(\alpha) &- (3\ell^2 + 6\ell)t(\alpha) = 0 \\
&\Downarrow \\
(2\ell^2 + 4\ell - 6 + \frac{18}{r(\alpha)})r(\alpha) &= (3\ell^2 + 6\ell)t(\alpha) \\
&\Downarrow \\
\frac{t(\alpha)}{r(\alpha)} &= \frac{2\ell^2 + 4\ell - 6 + \frac{18}{r(\alpha)}}{3\ell^2 + 6\ell}.
\end{aligned}
$$

As $\ell \to \infty$, we have

$$
\begin{aligned}
\frac{t(\alpha)}{r(\alpha)} \to \frac{2}{3} &\Rightarrow \frac{r(\alpha)}{3} - \frac{t(\alpha)}{2} \to 0 \\
&\Rightarrow \alpha \to 66.5022\ldots^\circ.
\end{aligned}
$$

- For $\ell = 6k + 2$, we have:

$$
\begin{aligned}
(12(\frac{\ell-2}{6})^2 + 12\frac{\ell-2}{6} + 2)r(\alpha) + 2 &- \ell(\frac{\ell}{2} + 1)t(\alpha) = 0 \\
&\Downarrow
\end{aligned}
$$

$$\dots \quad \text{(algebraic simplification)}$$

$$\Downarrow$$

$$(2\ell^2 + 4\ell - 4 + \frac{12}{r(\alpha)})r(\alpha) \quad = \quad (3\ell^2 + 6\ell)t(\alpha)$$

$$\Downarrow$$

$$\frac{t(\alpha)}{r(\alpha)} \quad = \quad \frac{2\ell^2 + 4\ell - 4 + \frac{12}{r(\alpha)}}{3\ell^2 + 6\ell}.$$

Again, as $\ell \to \infty$, we have

$$\frac{t(\alpha)}{r(\alpha)} \to \frac{2}{3} \quad \Rightarrow \quad \frac{r(\alpha)}{3} - \frac{t(\alpha)}{2} \to 0$$

$$\Rightarrow \quad \alpha \to 66.5022\dots^\circ.$$

- For $\ell = 6k + 4$, we have:

$$(12(\frac{\ell - 4}{6})^2 + 20\frac{\ell - 4}{6} + 7)r(\alpha) + 3 \quad - \quad \ell(\frac{\ell}{2} + 1)t(\alpha) = 0$$

$$\Downarrow$$

$$\dots \quad \text{(algebraic simplification)}$$

$$\Downarrow$$

$$(2\ell^2 + 4\ell - 6 + \frac{18}{r(\alpha)})r(\alpha) \quad = \quad (3\ell^2 + 6\ell)t(\alpha)$$

$$\Downarrow$$

$$\frac{t(\alpha)}{r(\alpha)} \quad = \quad \frac{2\ell^2 + 4\ell - 6 + \frac{18}{r(\alpha)}}{3\ell^2 + 6\ell}.$$

As $\ell \to \infty$, we again have

$$\frac{t(\alpha)}{r(\alpha)} \to \frac{2}{3} \quad \Rightarrow \quad \frac{r(\alpha)}{3} - \frac{t(\alpha)}{2} \to 0$$
$$\Rightarrow \quad \alpha \to 66.5022\ldots^\circ .$$

Therefore, α approaches $66.5022\ldots^\circ$ for each family of even-layered checkerboards as ℓ approaches infinity. □

We now consider ℓ odd.

Theorem 15 *For a generalized checkerboard with an odd number of layers, the crossover angle is either* $66.5022\ldots^\circ$ *or approaches* $66.5022\ldots^\circ$ *as* ℓ *approaches infinity.*

Proof: Consider each odd-layered checkerboard:

- For $\ell = 6k+1$, we have:

$$\begin{aligned}
(12(\frac{\ell-1}{6})^2 + 8\frac{\ell-1}{6} - 2)r(\alpha) + l(\alpha) &- 2((\frac{\ell-1}{2}+1)^2 - 1)t(\alpha) - r(\alpha) = 0 \\
&\Downarrow \\
&\ldots \quad \text{(algebraic simplification)} \\
&\Downarrow \\
(2\ell^2 + 4\ell - 24 + \frac{3l(\alpha)}{r(\alpha)})r(\alpha) &= (3\ell^2 + 6\ell - 9)t(\alpha) \\
&\Downarrow \\
\frac{t(\alpha)}{r(\alpha)} &= \frac{2\ell^2 + 4\ell - 24 + \frac{3l(\alpha)}{r(\alpha)}}{3\ell^2 + 6\ell - 9}.
\end{aligned}$$

Although we do not have an explicit expression for $l(\alpha)$, we know $l(\alpha)$ is fixed for a given α. Hence $l(\alpha)$ does not dominate ℓ as $\ell \to \infty$. So,

$$\begin{aligned} \frac{t(\alpha)}{r(\alpha)} \to \frac{2}{3} \quad &\Rightarrow \quad \frac{r(\alpha)}{3} - \frac{t(\alpha)}{2} \to 0 \\ &\Rightarrow \quad \alpha \to 66.5022\ldots^\circ . \end{aligned}$$

- For $\ell = 6k+3$, we have:

$$\begin{aligned} (12(\frac{\ell-3}{6})^2 + 16\frac{\ell-3}{6} + 2)r(\alpha) + l(\alpha) \quad &- \quad 2((\frac{\ell-1}{2}+1)^2 - 1)t(\alpha) - r(\alpha) = 0 \\ &\Downarrow \\ &\ldots \quad \text{(algebraic simplification)} \\ &\Downarrow \\ (2\ell^2 + 4\ell - 36 + \frac{6l(\alpha)}{r(\alpha)})r(\alpha) \quad &= \quad (3\ell^2 + 6\ell - 9)t(\alpha) \\ &\Downarrow \\ \frac{t(\alpha)}{r(\alpha)} \quad &= \quad \frac{2\ell^2 + 4\ell - 36 + \frac{6l(\alpha)}{r(\alpha)}}{3\ell^2 + 6\ell - 9}. \end{aligned}$$

Again, since $l(\alpha)$ is fixed for a particular α, $l(\alpha)$ does not dominate ℓ as $\ell \to \infty$. So,

$$\begin{aligned} \frac{t(\alpha)}{r(\alpha)} \to \frac{2}{3} \quad &\Rightarrow \quad \frac{r(\alpha)}{3} - \frac{t(\alpha)}{2} \to 0 \\ &\Rightarrow \quad \alpha \to 66.5022\ldots^\circ . \end{aligned}$$

- Finally, for $\ell = 6k+5$, we have:

$$\begin{aligned}
(12(\frac{\ell-5}{6}+1)^2-1)r(\alpha)+t(\alpha) &- 2((\frac{\ell-1}{2}+1)^2-1)t(\alpha)-r(\alpha)=0 \\
&\Downarrow \\
&\ldots \quad \text{(algebraic simplification)} \\
&\Downarrow \\
\frac{(\ell+1)^2-6}{3}r(\alpha) &- \frac{(\ell+1)^2-6}{2}t(\alpha)=0 \\
&\Downarrow \\
((\ell+1)^2-6)(\frac{r(\alpha)}{3} &- \frac{t(\alpha)}{2})=0.
\end{aligned}$$

Setting $(\ell+1)^2-6=0$ gives no integer values for ℓ, so we have no special cases to consider. Setting $\frac{r(\alpha)}{3}-\frac{t(\alpha)}{2}=0$ returns $\alpha = 66.5022\ldots°$, the same crossover established for the "powers of 2" grids.

Thus, the grid with $\ell = 6k+5$ has exactly the same crossover angle and the other two grids have a crossover angle that approaches $66.5022\ldots°$. □

Our work shows that the crossover angle α may not be constant in all cases but that it approaches $66.5022\ldots°$ in all cases. This implies the small angle of the rhombus (and *not* the size of the checkerboard) is the most important factor in choosing between the rhombus and triangle schemes. Thus, it is the underlying structure of the rhombus that most influences the structure of the shorter Steiner tree (and probably also the structure of the resulting Steiner minimal tree).

CHAPTER 7

FURTHER WORK

7.1 Generalized Ladders

In the $\ell = 6k + 1$ and the $\ell = 6k + 3$ cases for square grids, Chung, Gardner and Graham [2] make use of what they refer to as the Steiner tree over a *ladder* — that is, the Steiner tree over a row of four squares (see Figure 7.1). To actually calculate the length of the corresponding generalized trees, we need to develop a *generalized ladder* over four rhombuses.

In [3], Chung and Graham discuss the construction for $1 \times n$ ladders of squares (including the construction for the 1×4 ladder). Using the restrictions for ladders given in their work, as well as general Steiner point and Steiner tree restrictions, we have written *Mathematica* programs that allows us to construct their Steiner trees over square

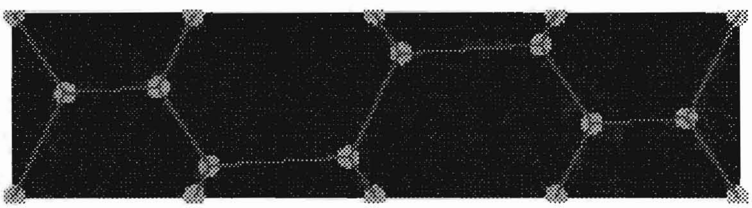


Figure 7.1 The Steiner tree over a 1×4 ladder of squares.

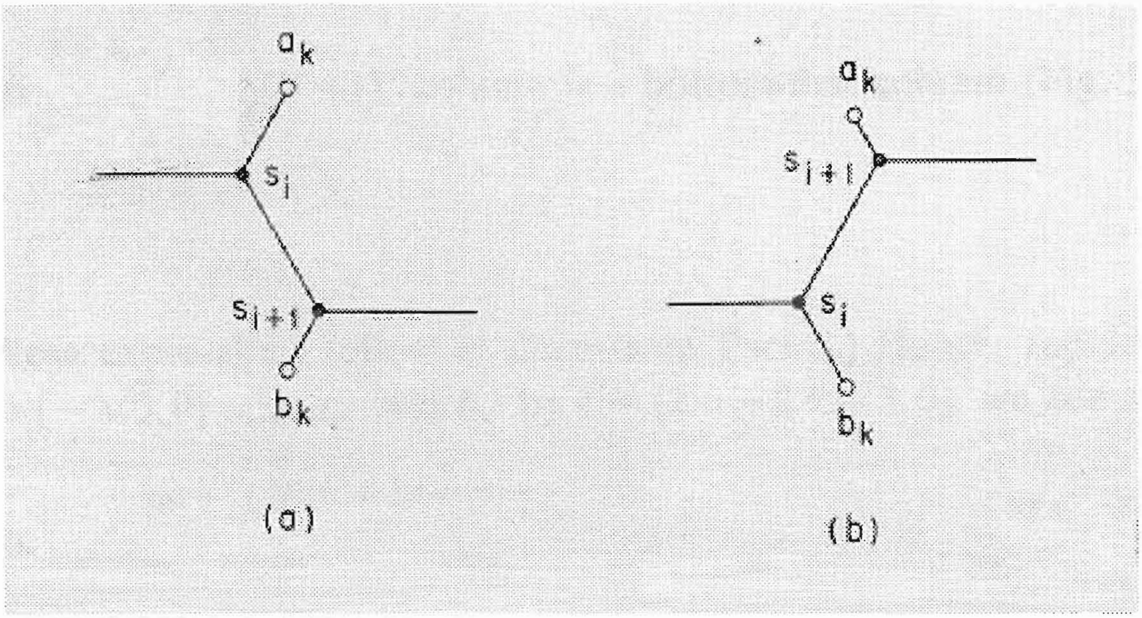


Figure 7.2 Top-first (a) and bottom-first (b) columns in a square ladder [3].

ladders. The algorithm involves building and solving simulaneously a series of equations (the number is dependent upon the size of the ladder) that return the (x, y) coordinates for the Steiner points in the ladder. (For more details, see Appendix C.)

Unfortunately, the ladder construction is difficult to generalize because the ladder restrictions that allow us to develop the series of equations in the first place are based on defining the optimal number of so-called bottom-first and top-first columns (see Figure 7.2). Due to the "slanted" nature of a rhombus, we have not yet been able to define "bottom-first" and "top-first" columns as they appear in [3]. Thus, a different technique is necessary for building rhombus ladders, leaving the ladder term in the $\ell = 6k + 1$ and the $\ell = 6k + 3$ cases technically "undefined" as it stands now.

We have developed experimental results for $\alpha = 60°$ using the bisection method on the slope of the line connecting the first two Steiner points. These results give us

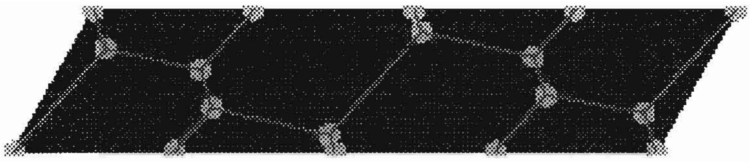


Figure 7.3 Our best Steiner tree over a 1×4 ladder of $60°$ rhombuses.

an idea of the length and structure of generalized ladder Steiner trees. (Appendix C contains the code used to build the 1×4 rhombus ladders based on this slope.) The best result we obtained for the 1×4 ladder is shown in Figure 7.3, which has a length of 8.18590760552377. (The minimum spanning tree for this ladder has a length of 9.)

The importance behind developing Steiner trees over rhombus ladders lies not only with finishing the generalization of checkerboards, but also with testing minimality. In [1], where they prove minimality of the square "powers of 2" grids, the authors do so by examining all other possible topologies including that of square ladders. It may be possible that a technique based on rhombus ladders, rather than on rhombuses or isosceles triangles, may return the Steiner minimal tree for some generalized checkerboards.

7.2 Other Types of Grids

Side-by-side with generalized checkerboards comes the question of looking at other types of "checkerboards". For example, Hwang and Du [14] looked at building Steiner trees over Chinese checkerboards. They used the equilateral triangle Steiner tree as the base unit, and developed two recursive schemes (one for the hexagon center and one for the triangle ends of the board) to fill the boards. Other common boards to

consider include checkerboards built from rectangles, parallelograms, and any other shape that can tile the plane. The problem of building Steiner trees over truly "generalized" checkerboards is indeed wide open.

LIST OF REFERENCES

[1] M. Brazil, T. Cole, J. Rubinstein, D. Thomas, J. Weng, and N. Wormald. Minimal Steiner trees for 2^kx2^k square lattices. *Journal of Combinatorial Theory, Series A*, 73:91–110, January 1996.

[2] F. Chung, M. Gardner, and R. Graham. Steiner trees on a checkerboard. *Mathematics Magazine*, 62(2):83–96, April 1989.

[3] F. Chung and R. Graham. Steiner trees for ladders. *Annals of Discrete Mathematics*, 2:173–200, 1978.

[4] R. Courant and H. Robbins. *What is Mathematics?* Oxford University Press, 1941.

[5] H. Coxeter. *Introduction to Geometry.* John Wiley, 1961.

[6] D. Du and F. Hwang. An approach for proving lower bounds: solution of Gilbert-Pollak's conjecture on the Steiner Ratio. In *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science*, 1990.

[7] D. Du and F. Hwang. A proof of Gilbert-Pollak's conjecture on the Steiner Ratio. *DIMACS Technical Report*, 1990.

[8] L. Foulds. *Graph Theory Applications.* Springer-Verlag, 1992.

[9] M. Gardner. Mathematical games: Casting a net on a checkerboard and other puzzles of the forest. *Scientific American*, 254:16–23, June 1986.

[10] M. Garey, R. Graham, and D. Johnson. The complexity of computing Steiner minimal trees. *SIAM J. of Appl. Math.*, 32:835–859, June 1977.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[12] E. Gilbert and H. Pollak. Steiner minimal trees. *SIAM J. of Appl. Math.*, 16:1–29, January 1968.

[13] R. Graham. Re: A question about steiner trees on a checkerboard. [Online] Available email: mvoelker@sun.iwu.edu from rlg@research.att.com, March 2, 1997.

[14] F. Hwang and D. Du. Steiner minimal trees on Chinese checkerboards. *Mathematics Magazine*, 64:332–339, December 1991. See also Mathematics Magazine 65 Discussion, October 1992.

[15] A. O. Ivanov and A. A. Tuzhilin. *Minimal Networks: The Steiner Problem and its Generalizations.* CRC Press, 1996.

[16] R. Johnsonbaugh. *Discrete Mathematics.* Macmillan Publishing, 1984.

[17] H. Pollak. Some remarks on the Steiner problem. *Journal of Combinatorial Theory, Series A*, 24:278–295, 1978.

[18] S. Salas and E. Hill. *Calculus: One and Several Variables.* John Wiley and Sons, sixth edition, 1990.

[19] R. J. Wilson. *Introduction to Graph Theory.* Longman Scientific and Technical, third edition, 1985.

APPENDIX A

GRAPHICS PROGRAMS

This appendix contains the *Mathematica* programs that draw and calculate the length of Steiner trees over rhombuses and isosceles triangles.

Program: Rhombus Steiner Tree

Given a rhombus specified by its small angle and lower left coordinate, this program calculates its Steiner tree, returning the length of the Steiner tree (as measured by the distance formula) and the corresponding graphics.

Initialization Cells

```
(* setting up notebook *)
  Off[General::spell]
  Needs["Graphics'Colors'"]

(* distance function *)
  Clear[Distance,x1,y1,x2,y2];
  Distance[{x1_, y1_}, {x2_, y2_}] =
           Sqrt[(x1-x2)^2 + (y1-y2)^2];

(* degree to radian conversion -- since Mathematica's
   default is radians *)
  Clear[d,DegRad];
  DegRad[d_] = (Pi/180) d;

(* finds center of rhombus: solve for intersection point
   between diagonals of rhombus *)
CenterRhom[{LLx_,LLy_},{ULx_,ULy_},{LRx_,LRy_},
           {URx_,URy_}] :=
Module[{x,diagonal1,diagonal2,cent,sol},
  diagonal1[x_] = (x-LRx) (LRy-ULy)/(LRx-ULx) + LRy;
  diagonal2[x_] = (x-LLx) (LLy-URy)/(LLx-URx) + LLy;
       sol = NSolve[diagonal1[x] == diagonal2[x],x];
       cent = {sol[[1,1,2]],diagonal1[sol[[1,1,2]]]};
  (* return center point *)
       cent
];

(* finds equilateral triangle *)
EquilTri[a_,{LLx_,LLy_},{ULx_,ULy_}] :=
Module[{b,pk,midpt1,midpt2,median1,median2,x,sol},
  (* calculate a's corresponding angle *)
       b = DegRad[120]-a;
  (* find peak of triangle *)
       pk = {N[LLx - Cos[b]], N[LLy + Sin[b]]};
  (* find centroid of triangle *)
       midpt1 = {N[LLx - 0.5 Cos[b]], N[LLy + 0.5 Sin[b]]};
       midpt2 = {N[LLx + 0.5 Cos[a]], N[LLy + 0.5 Sin[a]]};
       median1[x_] =
         (x-ULx) (ULy-midpt1[[2]])/(ULx-midpt1[[1]]) + ULy;
       median2[x_] =
         (x-pk[[1]]) (pk[[2]]-midpt2[[2]])/(pk[[1]]-midpt2[[1]]) +
            pk[[2]];
       sol = NSolve[median1[x] == median2[x],x];
       centroid = {sol[[1,1,2]], N[median1[sol[[1,1,2]]]]};
  (* return centroid and peak for further calculations *)
       {centroid,pk}
];

(* main module: calculates steiner points and length of
   steiner tree by taking in the lower left coordinate of the
```

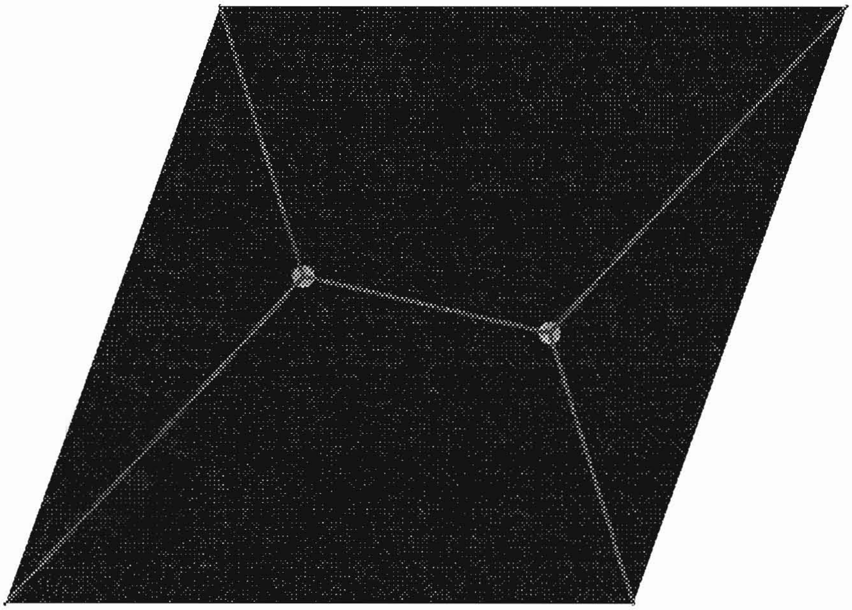
```
    rhombus and the measure of its small angle in degrees *)
Steiner[smangle_,LLx_,LLy_] :=
Module[{a,LL,UL,LR,UR,cent,centroid,pk,circ,s1,s2,stlength},
   (* define rhombus based on small angle and LL point *)
      a = DegRad[smangle];
      LL = {LLx,LLy};
      UL = {N[LLx + Cos[a]],N[LLy + Sin[a]]};
      LR = {N[LLx + 1],N[LLy]};
      UR = {N[UL[[1]] + 1],N[UL[[2]]]};
   (* calculate center for finding steiner tree *)
      cent = CenterRhom[LL,UL,LR,UR];
   (* calculate centroid and peak of equilateral triangle *)
      centroid = EquilTri[a,LL,UL][[1]];
      pk = EquilTri[a,LL,UL][[2]];
   (* find "equilateral" circle *)
      radius = N[Distance[centroid,LL]];
      circ = (x-centroid[[1]])^2 + (y-centroid[[2]])^2 ==
         radius^2;
   (* find steiner line *)
      line1 = y-cent[[2]] ==
       (x-cent[[1]]) (cent[[2]]-pk[[2]])/(cent[[1]]-pk[[1]]);
   (* find first steiner point *)
      sols = NSolve[{line1,circ},{x,y}];
      If[Chop[sols[[2,1,2]] - pk[[1]]] == 0,
            (* Chop replaces approximate real number with
               magnitude less than 10^(-10) by 0 *)
         {s1 = {sols[[1,1,2]], sols[[1,2,2]]}},
         {s1 = {sols[[2,1,2]], sols[[2,2,2]]}}];
   (* find second steiner point by reflection through center *)
      s2 = {cent[[1]]+(cent[[1]]-s1[[1]]),
            cent[[2]] - (s1[[2]]-cent[[2]])};
   (* calculate length of steiner tree using the distance
      formula *)
      stlength = N[Distance[UL,s1] + Distance[LL,s1] +
                   Distance[s1,s2]+ Distance[UR,s2] +
                   Distance[LR,s2]];
   (* return rhombus, steiner points and steiner tree length *)
      {LL,UL,LR,UR,s1,s2,stlength}
];


(* draws steiner tree for rhombus *)
StGraphics[LL_,UL_,LR_,UR_,s1_,s2_] =
Module[{rhombus,steiner1,steiner2,steinertree},
      rhombus = Graphics[{Blue,Polygon[{LL,UL,UR,LR}]}];
      steiner1 = Graphics[{Green,PointSize[0.025],Point[s1]}];
      steiner2 = Graphics[{Green,PointSize[0.025],Point[s2]}];
      steinertree = {Graphics[{Red,Line[{UL,s1}]}],
                     Graphics[{Red,Line[{LL,s1}]}],
                     Graphics[{Red,Line[{s1,s2}]}],
                     Graphics[{Red,Line[{UR,s2}]}],
                     Graphics[{Red,Line[{LR,s2}]}]};
      Show[rhombus,steiner1,steiner2,steinertree,
           AspectRatio->Automatic,
           DisplayFunction->Identity,
           PlotRange->All]
];
```

■ Test1

```
Clear[smangle,LLx,LLy]
smangle = 70;
LLx = 0;
LLy = 0;
Steiner[smangle,LLx,LLy]
```

```
{{0, 0}, {0.3420201433256688, 0.939692620785908}, {1., 0},
 {1.342020143325669, 0.939692620785908}, {0.4745226100533893, 0.5141447743490016},
 {0.86749753327228, 0.4255478464369067}, 2.693546124609247}
```

```
Show[StGraphics[Steiner[smangle,LLx,LLy][[1]],
                Steiner[smangle,LLx,LLy][[2]],
                Steiner[smangle,LLx,LLy][[3]],
                Steiner[smangle,LLx,LLy][[4]],
                Steiner[smangle,LLx,LLy][[5]],
                Steiner[smangle,LLx,LLy][[6]]],
     DisplayFunction->$DisplayFunction];
```

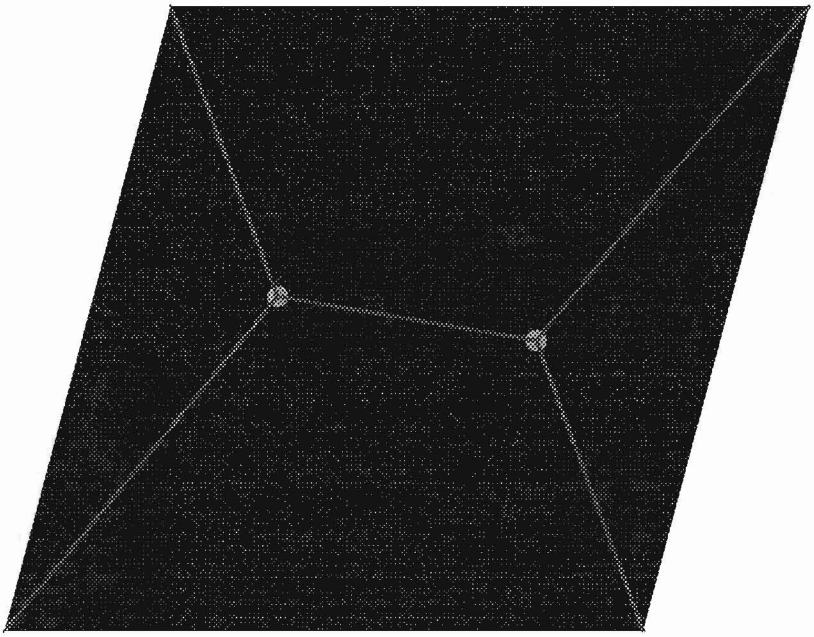


■ Test2

```
Clear[smangle,LLx,LLy]
smangle = 75;
LLx = 0;
LLy = 0;
Steiner[smangle,LLx,LLy]
```

```
{{0, 0}, {0.2588190451025206, 0.965925826289068}, {1., 0},
 {1.258819045102521, 0.965925826289068}, {0.4264858900897018, 0.5169947373503931},
 {0.832333155012819, 0.4489310889386749}, 2.710362561531433}
```

```
Show[StGraphics[Steiner[smangle,LLx,LLy][[1]],
                Steiner[smangle,LLx,LLy][[2]],
                Steiner[smangle,LLx,LLy][[3]],
                Steiner[smangle,LLx,LLy][[4]],
                Steiner[smangle,LLx,LLy][[5]],
                Steiner[smangle,LLx,LLy][[6]]],
     DisplayFunction->$DisplayFunction];
```

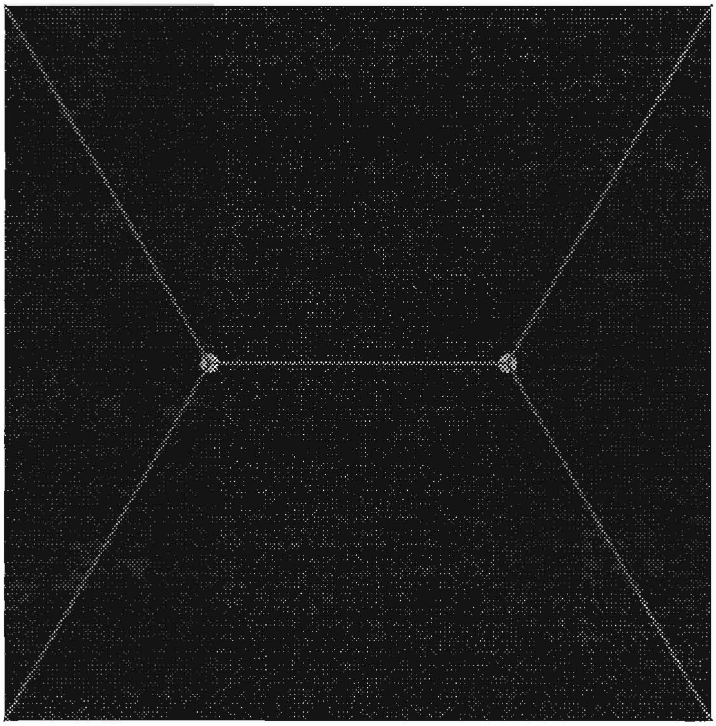


■ Test3

```
Clear[smangle,LLx,LLy]
smangle = 90;
LLx = 0;
LLy = 0;
Steiner[smangle,LLx,LLy]
```

```
{{0, 0}, {0, 1.}, {1., 0}, {1., 1.}, {0.288675134594813, 0.5}, {0.711324865405187, 0.5},
 2.732050807568877}
```

```
Show[StGraphics[Steiner[smangle,LLx,LLy][[1]],
                Steiner[smangle,LLx,LLy][[2]],
                Steiner[smangle,LLx,LLy][[3]],
                Steiner[smangle,LLx,LLy][[4]],
                Steiner[smangle,LLx,LLy][[5]],
                Steiner[smangle,LLx,LLy][[6]]],
     DisplayFunction->$DisplayFunction];
```

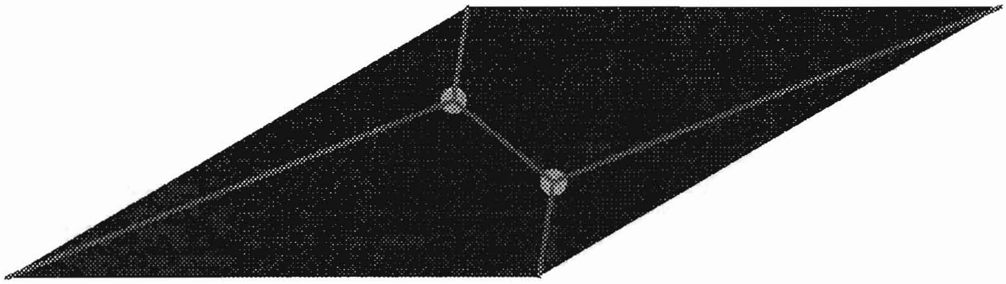


■ Test4

```
Clear[smangle,LLx,LLy]
smangle = 30;
LLx = 0;
LLy = 0;
Steiner[smangle,LLx,LLy]
```

```
{{0, 0}, {0.866025403784439, 0.5}, {1., 0}, {1.866025403784438, 0.5},
 {0.83903671447014, 0.3255423698129907}, {1.026988689314298, 0.1744576301870098},
 2.394170170971328}
```

```
Show[StGraphics[Steiner[smangle,LLx,LLy][[1]],
                Steiner[smangle,LLx,LLy][[2]],
                Steiner[smangle,LLx,LLy][[3]],
                Steiner[smangle,LLx,LLy][[4]],
                Steiner[smangle,LLx,LLy][[5]],
                Steiner[smangle,LLx,LLy][[6]]],
     DisplayFunction->$DisplayFunction];
```

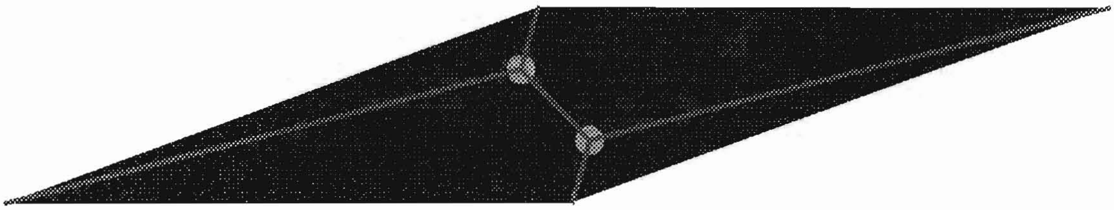


■ Test5

```
Clear[smangle,LLx,LLy]
smangle = 20;
LLx = 0;
LLy = 0;
Steiner[smangle,LLx,LLy]
```

```
{{0, 0}, {0.939692620785909, 0.3420201433256687}, {1., 0},
 {1.939692620785908, 0.3420201433256687}, {0.909199100618674, 0.232997856695233},
 {1.030493520167234, 0.1090222866304356}, 2.277013950529091}
```

```
Show[StGraphics[Steiner[smangle,LLx,LLy][[1]],
                Steiner[smangle,LLx,LLy][[2]],
                Steiner[smangle,LLx,LLy][[3]],
                Steiner[smangle,LLx,LLy][[4]],
                Steiner[smangle,LLx,LLy][[5]],
                Steiner[smangle,LLx,LLy][[6]]],
     DisplayFunction->$DisplayFunction];
```



Steiner Trees Over Isosceles Triangles

This notebook draws and calculates the length of a Steiner tree (as measured by the distance formula) over an isosceles triangle (taken from a rhombus).

Initialization Cells

```
(* set up notebook *)
Off[General::spell]
Needs["Graphics‘Colors‘"]


(* convert angle given in degrees to radians *)
Clear[d,DegRad];
DegRad[d_] = (Pi/180) d;


(* calculate the length of steiner tree and coordinates of
   steiner point for triangle by taking in the 3 vertices of
   the triangle and the measure of the angle between the two
   congruent sides in degrees *)
TriTree[angle_,{apexx_,apexy_},{v1x_,v1y_},{v2x_,v2y_}] :=
Module[{a,b,s,t,length,stedge,deltax,deltay,steiner,x,y},
    (* convert a to radians *)
       a = DegRad[angle];
    (* find corresponding base angle *)
       b = (DegRad[180]-a)/2;
    (* find vertex that has same y-coordinate as apex (for
       calculating coordinates of steiner point) - possible
       because triangle is part of rhombus *)
       If[Chop[N[v1y-apexy]] == 0,
           {x = v1x; y = v1y},
           {x = v2x; y = v2y}];
    (* calculate length of congruent sides *)
       s = Abs[x-apexx];
    (* calculate length of side opposite apex *)
       t = 2 s Cos[b];
    (* calculate length of steiner tree *)
       length = N[s Sin[b] + (Sqrt[3]/2) t];

    (* find coordinates of steiner point: *)
      (* find distance between steiner point and v1 (or v2) *)
       stedge = (1/2) t/Sin[DegRad[60]];
      (* calculate change in x and y from v1 to steiner
       point *)
       deltax = Cos[b-DegRad[30]] stedge;
       deltay = Sin[b-DegRad[30]] stedge;
      (* determine if steiner point located to right or to the
       left of apex and give steiner point coordinates *)
       If[N[apexx] < N[x],
           steiner = {N[x-deltax],N[y+deltay]},
           steiner = {N[x+deltax],N[y-deltay]}];

    (* return steiner tree length and coordinates of steiner
       point *)
       {length,steiner}
];

(* draw triangle and corresponding steiner tree *)
TriGraphics[apex_,v1_,v2_,steiner_] :=
Module[{stpt,triangle,sttree},
  (* steiner point: *)
    stpt = Graphics[{Green,PointSize[0.025],Point[steiner]}];
  (* triangle: *)
    triangle = Graphics[{Blue,Polygon[{apex,v1,v2}]}];
  (* steiner tree: *)
    sttree = {Graphics[{Red,Line[{apex,steiner}]}],
              Graphics[{Red,Line[{v1,steiner}]}],
              Graphics[{Red,Line[{v2,steiner}]}]};

  (* set up graphics *)
    Show[triangle,stpt,sttree,
         AspectRatio->Automatic,
         DisplayFunction->Identity]
];
```

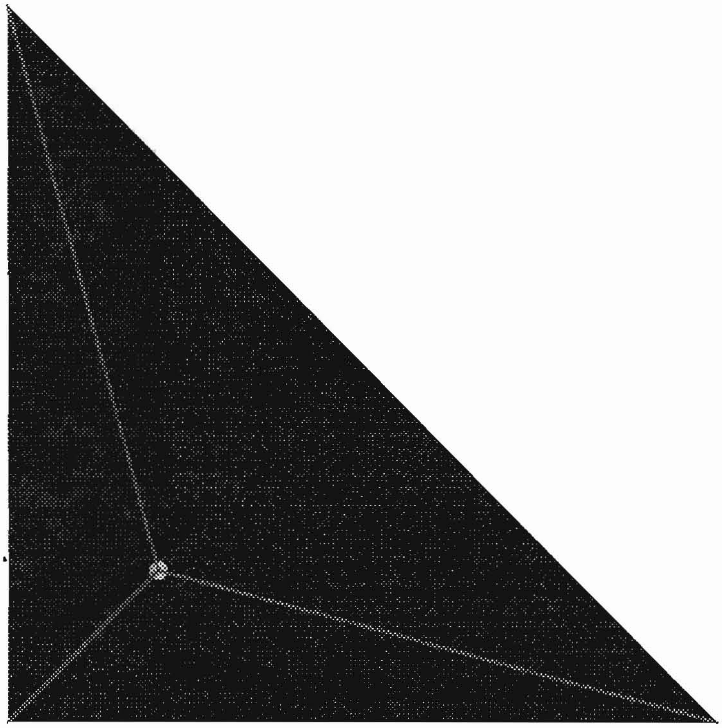
■ Test 1

```
Clear[angle,apex,vertex1,vertex2,tree]
angle = 90;
apex = {0,0};
vertex1 = {1,0};
vertex2 = {Cos[DegRad[angle]],Sin[DegRad[angle]]};
```

```
tree = TriTree[angle,apex,vertex2,vertex1]
```

```
{1.931851652578136, {0.2113248654051871, 0.211324865405187}}
```

```
Show[TriGraphics[apex,vertex1,vertex2,tree[[2]]],
     DisplayFunction->$DisplayFunction];
```



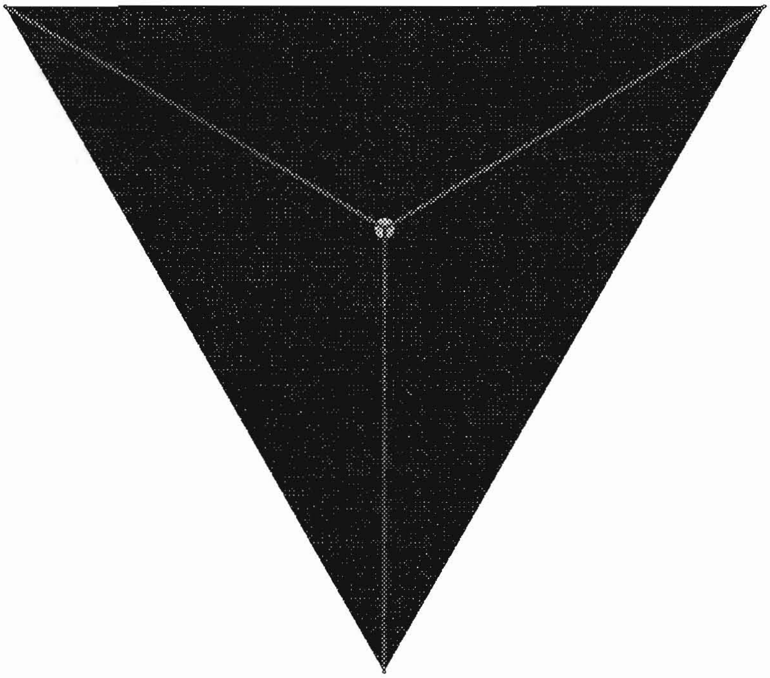
■ Test 2

```
Clear[angle,apex,vertex1,vertex2,tree]
angle = 60;
apex = {1,1};
vertex1 = {0,1};
vertex2 = {1-Cos[DegRad[angle]],1-Sin[DegRad[angle]]};
```

```
tree = TriTree[angle,apex,vertex1,vertex2]
```

```
{1.732050807568877, {0.5, 0.7113248654051871}}
```

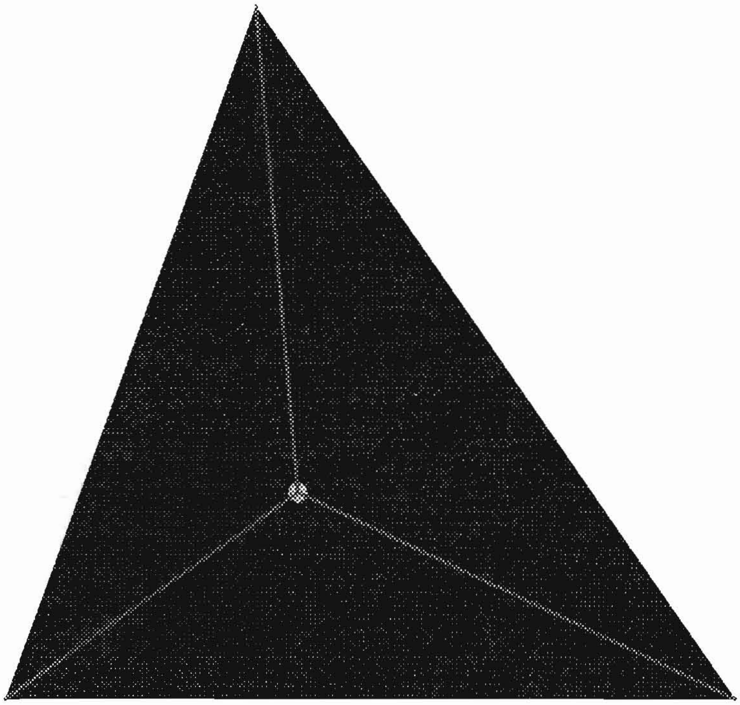
```
Show[TriGraphics[apex,vertex1,vertex2,tree[[2]]],
     DisplayFunction->$DisplayFunction];
```



■ Test 3

```
Clear[angle,apex,vertex1,vertex2,tree]
angle = 70;
apex = {0,0};
vertex1 = {1,0};
vertex2 = {Cos[DegRad[angle]],Sin[DegRad[angle]]};

tree = TriTree[angle,apex,vertex1,vertex2]

{1.8126155740733, {0.3997441778797097, 0.2799038867068159}}
```

```
Show[TriGraphics[apex,vertex1,vertex2,tree[[2]]],
     DisplayFunction->$DisplayFunction];
```

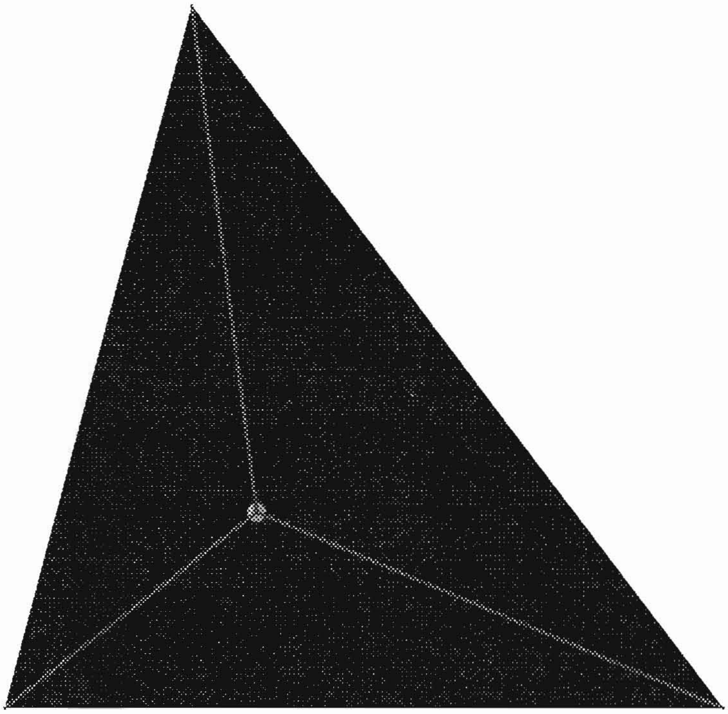


■ Test 4

```
Clear[angle,apex,vertex1,vertex2,tree]
angle = 75;
apex = {0,0};
vertex1 = {1,0};
vertex2 = {Cos[DegRad[angle]],Sin[DegRad[angle]]};

tree = TriTree[angle,apex,vertex2,vertex1]
```

```
{1.847759065022573, {0.3505707546386576, 0.2690024012303924}}
```

```
Show[TriGraphics[apex,vertex1,vertex2,tree[[2]]],
     DisplayFunction->$DisplayFunction];
```



APPENDIX B

NUMERIC PROGRAMS

This appendix contains the *Mathematica* programs that return numerical data used in this paper.

Steiner Trees for Rhombuses and Triangles

■ Rhombus Steiner Tree

In this section, we find an equation for the length of the rhombus Steiner tree over a unit rhombus in terms of the small angle, a (which we assume is in radians). In order to do this, we use a modified version of the rhombus algorithm developed before: we assume that the rhombus' left side is parallel to the y–axis, while its base forms some angle with the x–axis. This allows us to define the peak of the equilateral triangle as constant. The center coordinates of the rhombus will vary with the small angle.

In general, we find the length of the Steiner tree over one of the triangles used to build the final Steiner tree for the rhombus. After this, we multiply this value by two in order to cover the other triangle.

To find the length of the Steiner tree over one of the triangles, we use Gilbert and Pollak's simplified algorithm (which makes use of Coexter's work). With this theorem, we only need to find the distance between the center of our rhombus (representing the third vertex of the triangle over which we are drawing the Steiner tree) and the peak of the equilateral triangle.

Following the definition of the equation, we verify that it gives correct results for test angles.

Initialization Cells

```
(* setting up notebook *)
  Off[General::spell]
  Needs["Graphics'Colors'"]

(* distance function *)
  Clear[Distance,x1,y1,x2,y2];
  Distance[{x1_, y1_}, {x2_, y2_}] =
           Sqrt[(x1-x2)^2 + (y1-y2)^2];

(* degree to radian conversion -- since Mathematica's
   default is radians *)
  Clear[d,DegRad];
  DegRad[d_] = (Pi/180) d;

(* finds center of rhombus: solve for intersection point
   between diagonals of rhombus *)
CenterRhom[{LLx_,LLy_},{ULx_,ULy_},{LRx_,LRy_},
           {URx_,URy_}] :=
Module[{x,diagonal1,diagonal2,cent,sol},
  diagonal1[x_] = (x-LRx) (LRy-ULy)/(LRx-ULx) + LRy;
  diagonal2[x_] = (x-LLx) (LLy-URy)/(LLx-URx) + LLy;
       sol = Solve[diagonal1[x] == diagonal2[x],x];
       cent = {sol[[1,1,2]],diagonal1[sol[[1,1,2]]]};
  (* return center point *)
       cent
];
```

Finding the rhombus formula:

1. Set up rhombus coordinates: a is the measure of the small angle of the rhombus in radians

```
Clear[LL,UL,LR,UR,a]
LL = {0,0};
UL = {0,1};
LR = {Cos[Pi/2-a],Sin[Pi/2-a]};
UR = {Cos[Pi/2-a],1+Sin[Pi/2-a]};
```

2. Calculate center of rhombus:

```
Clear[cent]
cent = Simplify[CenterRhom[LL,UL,LR,UR]]
```

$$\left\{\frac{\mathrm{Sin}[a]}{2},\ \mathrm{Cos}\left[\frac{a}{2}\right]^2\right\}$$

3. Calculate peak of equilateral triangle:

```
Clear[pk]
pk = {-Sqrt[3]/2, 1/2}
```

$$\left\{-\frac{\sqrt{3}}{2},\ \frac{1}{2}\right\}$$

4. Find distance between peak and center (gives the length of the Steiner tree over the first triangle):

```
Clear[firsttri]
firsttri = Distance[cent,pk]
```

$$\sqrt{\left(-\frac{1}{2}+\text{Cos}\left[\frac{a}{2}\right]^2\right)^2+\left(\frac{\sqrt{3}}{2}+\frac{\text{Sin}[a]}{2}\right)^2}$$

5. Find distance of Steiner tree (multiply length found above by 2 since build Steiner tree over two congruent triangles and simplify using trig properties):

```
rhombus[a_] = Simplify[2 firsttri,Trig->True]
```

$$\sqrt{4+2\sqrt{3}\ \text{Sin}[a]}$$

Test Cases:

```
N[rhombus[DegRad[60]]]
```

2.645751311064591

```
N[rhombus[DegRad[90]]]
```

2.732050807568877

```
N[rhombus[DegRad[70]]]
```

2.693546124609247

```
N[rhombus[DegRad[75]]]
```

2.710362561531432

These values correspond to the ones obtained from the original rhombus program.

■ Isosceles Triangle Steiner Tree

The isosceles triangle steiner tree formula is much easier to find -- the fact that two of the sides are congruent lets us simplify the problem tremendously.

In this case, a represents the angle at the peak of the triangle (opposite the single, (possibly) non-congruent side) and is given in radians. The angle b represents the base angles of the triangle (opposite each of the two congruent sides). The length s represents the length of one of the congruent sides, which we take to be equal to 1 (since we are using a unit rhombus), and the length t represents the length of the side opposite the peak.

The formula is given in triangle[a].

Finding the triangle formula:

1. Setup for the formula:

```
Clear[a,b,s,t]
b = (DegRad[180]-a)/2
s = 1
t = 2 s Cos[b]
```

$$\frac{1}{2}(-a+\pi)$$

1

$$2\,\mathrm{Cos}\left[\frac{1}{2}(-a+\pi)\right]$$

2. The isosceles triangle formula:

```
Clear[triangle]
triangle[a_] = s Sin[b] + (Sqrt[3]/2) t
```

$$\sqrt{3}\,\mathrm{Cos}\left[\frac{1}{2}(-a+\pi)\right]+\mathrm{Sin}\left[\frac{1}{2}(-a+\pi)\right]$$

(This is not the only version of the triangle formula we can use properties of sines and cosines to put into other, possibly easier-to-use forms when necessary.)

Test Cases:

```
N[triangle[DegRad[60]]]
```

1.732050807568877

```
N[triangle[DegRad[90]]]
```

1.931851652578136

```
N[triangle[DegRad[70]]]
```

1.8126155740733

```
N[triangle[DegRad[75]]]
```

1.847759065022573

Comparing Full Rhombus Steiner Tree to Other Obvious Case

Notebook to examine the relationship of (tritree[a]+1) to (rhomtree[a]) for 0 < a < 60 degrees.

■ Initialization

```
Off[General::"spell"]
Needs["Graphics`Colors`"]

(* conversion from degrees to radians *)
Clear[d, DegRad];
```

$$\text{DegRad[d\_] := }\frac{\pi\,\text{d}}{180};$$

```
(* formulas for rhombus and triangle Steiner trees -- both
take in a in degrees and automatically convert to
the default of radians *)
Clear[rhomtree];
```

$$\text{rhomtree[a\_] := }\sqrt{4+2\sqrt{3}\ \text{Sin[DegRad[a]]}};$$

```
Clear[tritree];
```

$$\text{tritree[a\_] := 2 Cos}\left[\frac{\text{DegRad[a]}}{2}-\text{DegRad[60]}\right];$$

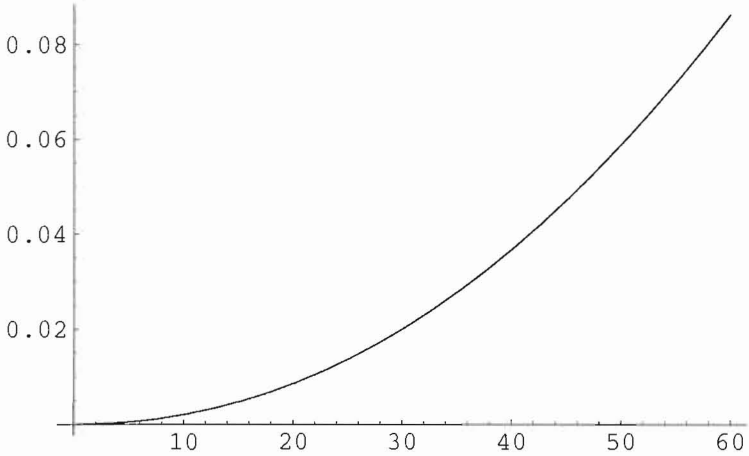
■ The Test

Let f[a] represent the difference between (tritree[a]+1) and (rhomtree[a]). To show that rhomtree[a] is better, we want to show that f[a] > 0 for a > 0.

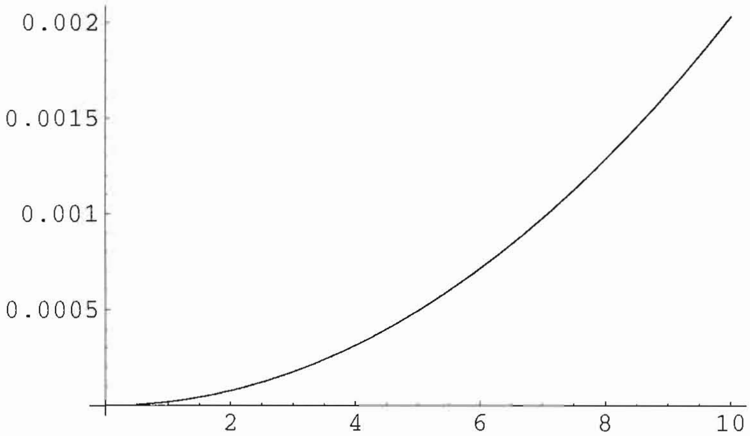
```
Clear[a, f]
f[a_] = tritree[a] + 1 - rhomtree[a]
```

$$1+2\,\text{Cos}\left[\frac{\pi}{3}-\frac{a\,\pi}{360}\right]-\sqrt{4+2\sqrt{3}\ \text{Sin}\left[\frac{a\,\pi}{180}\right]}$$

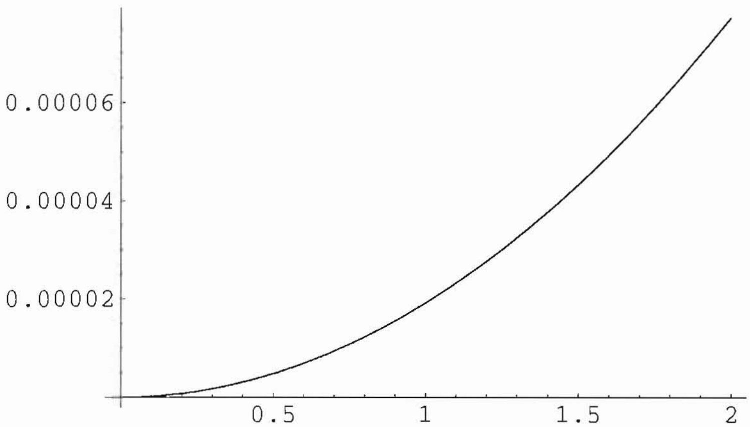
```
Plot[f[a], {a, 0, 60}];
```



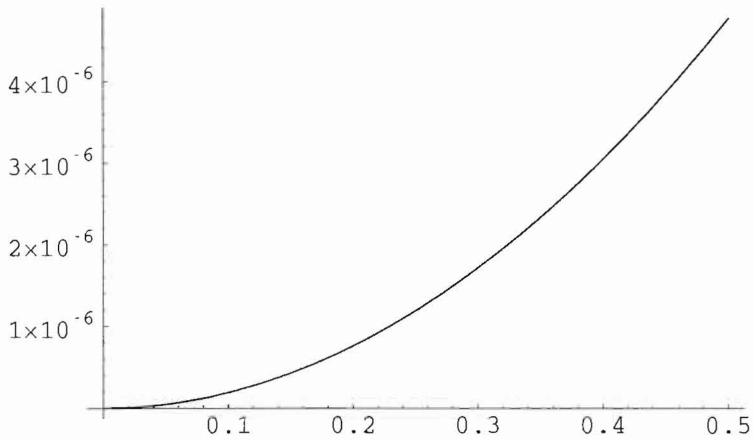

```
Plot[f[a], {a, 0, 10}];
```




```
Plot[f[a], {a, 0, 2}];
```



```
Plot[f[a], {a, 0, 0.5}];
```



f(a) appears to be greater than zero until very small values of a are reached. We know that it is continuous (since we have no discontinuous points by definition of f). What happens to f(a) around a = 0?

```
Limit[f[a], a -> 0]
```

0

```
f[0]
```

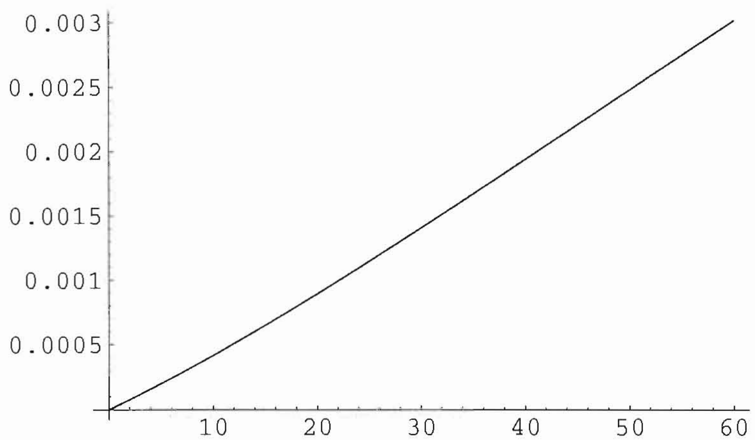
0

It appears that f[a] > 0 for a > 0. We can verify this by considering f'[a]:

```
f'[a]
```

$$-\frac{\pi \operatorname{Cos}\left[\frac{a\pi}{180}\right]}{60\sqrt{3}\sqrt{4+2\sqrt{3}\operatorname{Sin}\left[\frac{a\pi}{180}\right]}} + \frac{1}{180}\pi \operatorname{Sin}\left[\frac{\pi}{3} - \frac{a\pi}{360}\right]$$

```
Plot[f'[a], {a, 0, 60}];
```



```
FindRoot[f'[a] == 0, {a, 0}]

{a → 0.}
```

It appears that f'[a] is positive for a > 0. Algebraically, we have:

$$-\frac{\sqrt{3}\ \mathrm{Cos}[a]}{\sqrt{4+2\sqrt{3}\ \mathrm{Sin}[a]}}<0$$

and

Sin[60–a/2] > 0

since 0 < a < 60 degrees. The minimum of the negative term, i.e. the maximum negative value, occurs at a = 0 with a value of

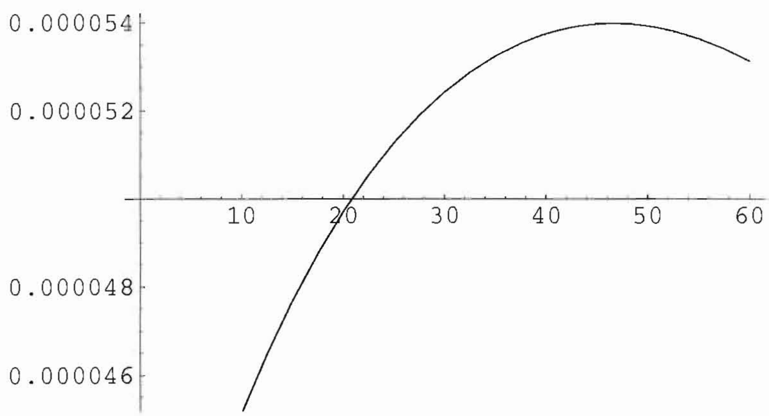
```
N[-Sqrt[3] Cos[0] / Sqrt[4 + 0]]

-0.866025
```

At a = 0, the positive term is

```
N[Sin[DegRad[60 - 0]]]

0.866025
```

So the two terms cancel out at a = 0. However, for a > 0 (a < 60 degrees), the negative term increases towards 0, i.e. it becomes less negative; the positive term increases also. This implies that f'[a] will be positive for 0 < a < 60 degrees. This is also verified by looking at f''[a]:

```
Plot[f''[a], {a, 0, 60}];
```



Since f'[a] is continuous on our interval and f''[a] is positive, f'[a] is strictly increasing. So, f'[a] > 0 for 0 <a < 60 degrees. This implies that f[a] is strictly increasing on our interval, and so f[a] > 0 on our interval! Therefore, rhomtree[a] < (tritree[a]+1) for
0 < a < 60 degrees.

Comparison of "Powers of 2" Grids

Program to compare the crossover points found on other girds with number of points = power of 2.

■ Initialization

```
(* degree-to-radian conversion *)
Clear[d,DegRad];
DegRad[d_] := (Pi/180) d;

(* length of rhombus steiner tree *)
Clear[rhomtree];
rhomtree[a_] := 2*((-1/2 + Cos[a/2]^2)^2 + (3^(1/2)/2 +
                Sin[a]/2)^2)^(1/2);

(* length of triangle steiner tree *)
Clear[tritree];
tritree[a_] := Sin[(DegRad[180]-a)/2] +
               Sqrt[3] Cos[(DegRad[180]-a)/2];

(* number of rhombuses for grid *)
Clear[numrhom];
numrhom[layers_] := (4^(Log[layers+1]/Log[2]) - 1) / 3;

(* number of triangles for grid *)
numtri[layers_] :=
Module[{k,ntri,nrhom},
       If[Mod[layers,2] == 0,
          {k = layers/2; ntri = 2 k (k+1); nrhom = 0},
             (* even number of layers *)
          {k = (layers-1)/2; ntri = 2 ((k+1)^2-1); nrhom = 1}
             (* odd number of layers *)
       ];
       {ntri,nrhom}
];

(* length of rhombus grid tree *)
Clear[rhomgrid]
rhomgrid[a_,layers_] := numrhom[layers] rhomtree[a];

(* length of triangle grid tree *)
Clear[trigrid];
trigrid[a_,layers_] := numtri[layers][[1]] tritree[a] +
                       numtri[layers][[2]] rhomtree[a];
```

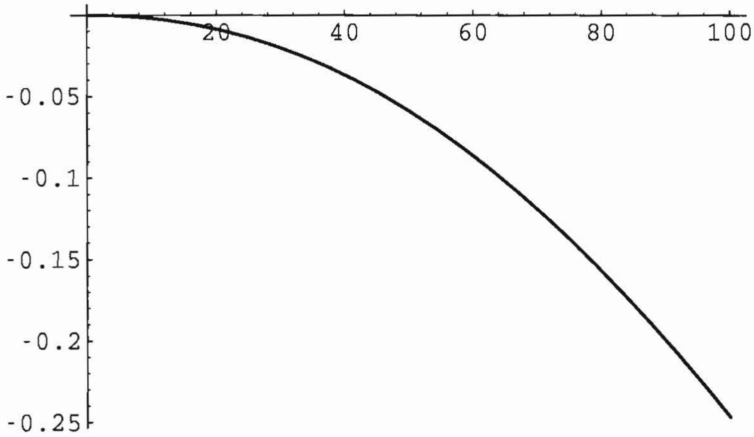
■ 1x1 Grid

We will use the "triangle tree" as one triangle, built in the usual fashion, pull one edge. The rhombus tree will be built as usual. We will not have to specify the number of layers (since there is only one), and therefore we can directly use rhomtree and tritree.

```
Clear[a,tri]
tri[a_] = tritree[a] + 1
```

$$1 + \sqrt{3}\,\text{Cos}\left[\frac{1}{2}\,(-a + \pi)\right] + \text{Sin}\left[\frac{1}{2}\,(-a + \pi)\right]$$

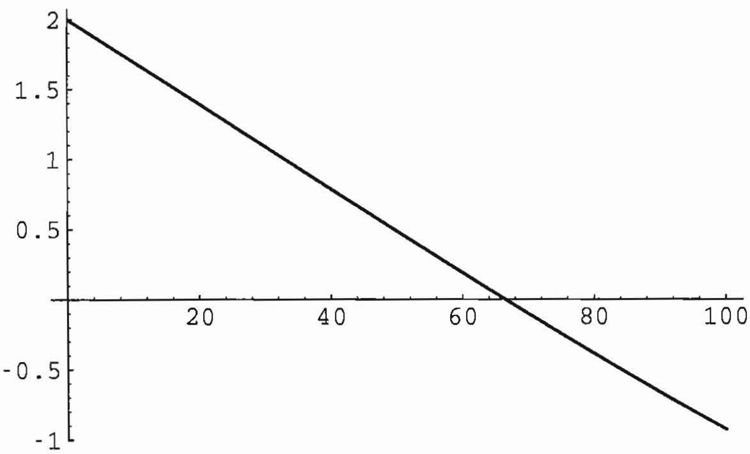
```
Clear[grid1]
grid1 = Plot[rhomtree[DegRad[a]]-tri[DegRad[a]],
             {a,0,100},
             PlotStyle->{Black}];
```



As we get farther from 0 Degrees, the length of the "triangle tree"over a single rhombus gets larger than the length of the rhombus tree. Thus, this supports the use of the rhombus tree for a 1x1 grid.

■ 3x3 Grid

```
Clear[layers,a,grid3]
layers = 3;
grid3 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,0,100},
        PlotStyle->{Blue}];
```

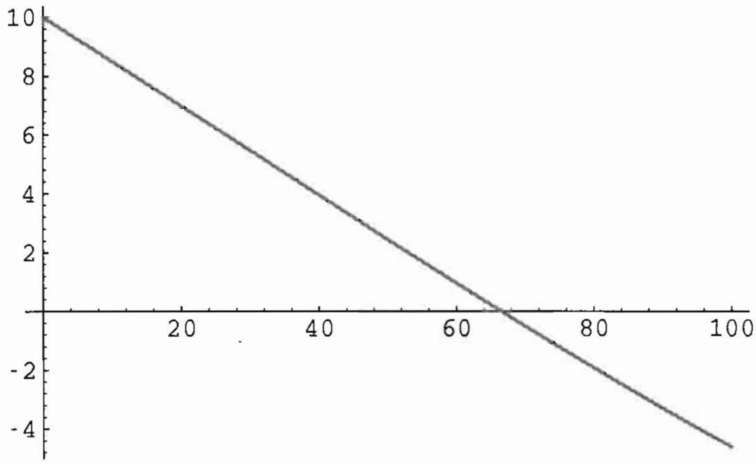


```
FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}]
```

{a → 66.50222270248989}

■ 7x7 Grid

```
Clear[layers,a,grid7]
layers = 7;
grid7 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,0,100},
        PlotStyle->{Green}];
```

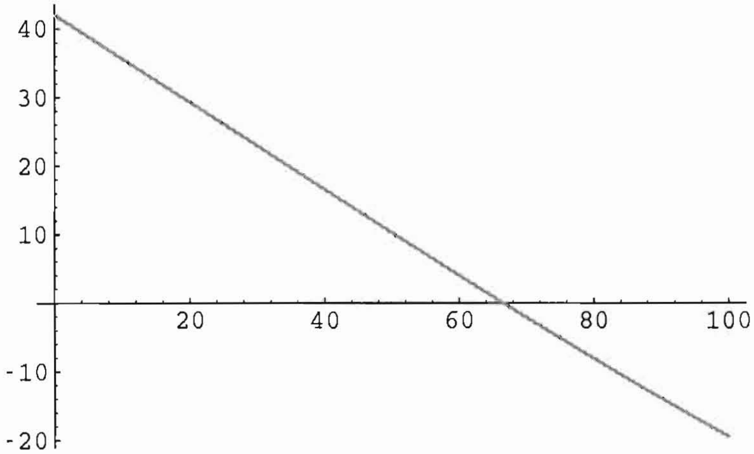


```
FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}]
```

{a → 66.50222270248958}

■ 15x15 Grid

```
Clear[layers,a,grid15]
layers = 15;
grid15 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,0,100},
        PlotStyle->{Cyan}];
```

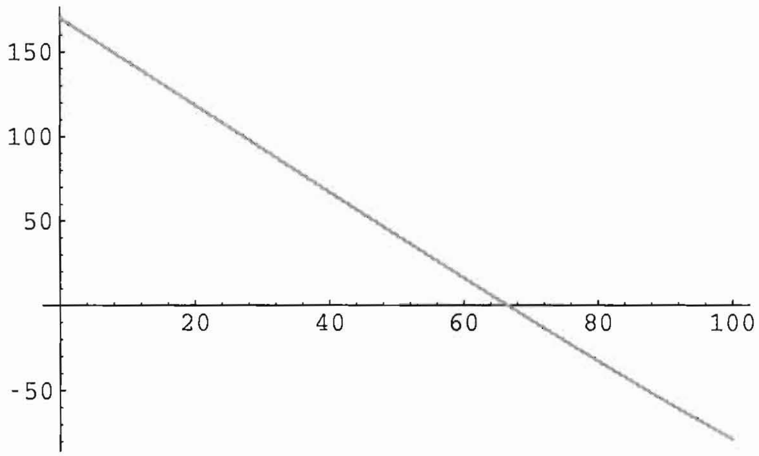


```
FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}]
```

{a → 66.50222270248991}

■ 31x31 Grid

```
Clear[layers,a,grid31]
layers = 31;
grid31 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,0,100},
        PlotStyle->{Banana}];
```

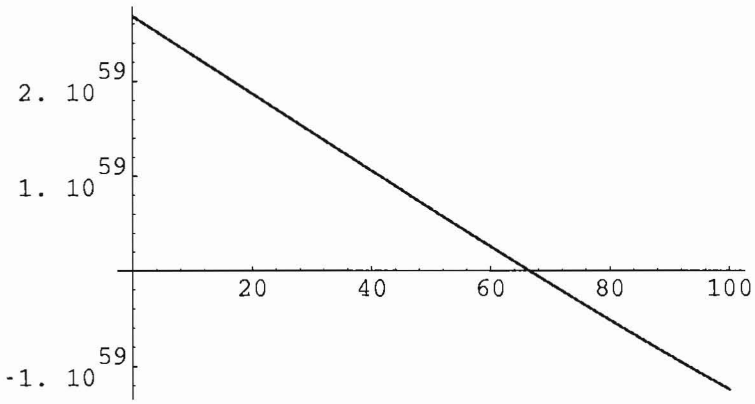


```
FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}]
```

{a → 66.50222270248985}

■ (2^100–1)x(2^100–1) Grid

```
Clear[layers,a,grid100]
layers = 2^100-1;
grid100 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,0,100},
        PlotStyle->{Red}];
```

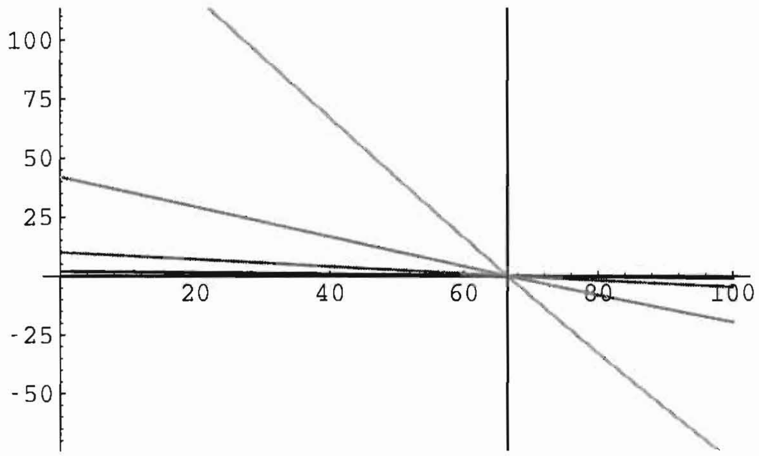


```
FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}]
```

{a → 66.50222270248985}

■ (Graph) Comparison

```
Show[grid1,grid3,grid7,grid15,grid31,grid100];
```



It appears that each grid has the same crossover point -- the difference between the sizes of the rhombus-based tree and the triangle-based tree increase faster on larger grids, but the crossover point still remains the same!

Comparison of "Powers of 2" Grids: Zoom–In on Interval from 66 Degrees to 67 Degrees

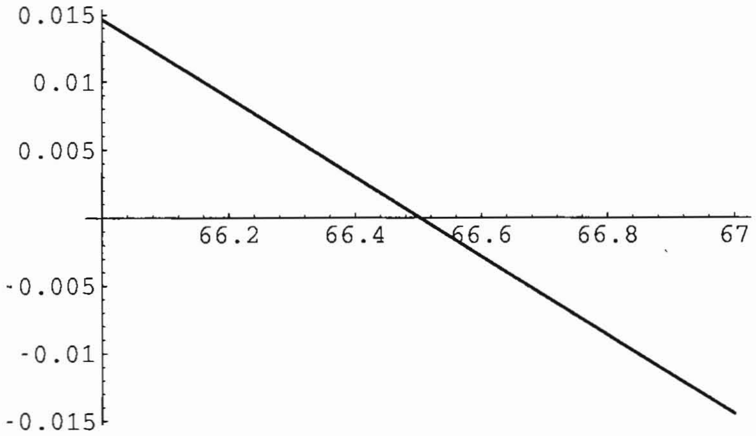
Program to compare the crossover points found on other girds with number of points = power of 2.

■ Initialization

```
(* set up notebook *)
Off[General::spell]
Needs["Graphics'Colors'"]

(* degree-to-radian conversion *)
Clear[d,DegRad];
DegRad[d_] := (Pi/180) d;

(* length of rhombus steiner tree *)
Clear[rhomtree];
rhomtree[a_] := 2*((-1/2 + Cos[a/2]^2)^2 + (3^(1/2)/2 +
                 Sin[a]/2)^2)^(1/2);

(* length of triangle steiner tree *)
Clear[tritree];
tritree[a_] := Sin[(DegRad[180]-a)/2] +
              Sqrt[3] Cos[(DegRad[180]-a)/2];

(* number of rhombuses for grid *)
Clear[numrhom];
numrhom[layers_] := (4^(Log[layers+1]/Log[2]) - 1) / 3;

(* number of triangles for grid *)
numtri[layers_] :=
Module[{k,ntri,nrhom},
       If[Mod[layers,2] == 0,
          {k = layers/2; ntri = 2 k (k+1); nrhom = 0},
             (* even number of layers *)
          {k = (layers-1)/2; ntri = 2 ((k+1)^2-1); nrhom = 1}
             (* odd number of layers *)
       ];
       {ntri,nrhom}
];

(* length of rhombus grid tree *)
Clear[rhomgrid]
rhomgrid[a_,layers_] := numrhom[layers] rhomtree[a];

(* length of triangle grid tree *)
Clear[trigrid];
trigrid[a_,layers_] := numtri[layers][[1]] tritree[a] +
                        numtri[layers][[2]] rhomtree[a];
```

■ 3x3 Grid

```
Clear[layers,a,grid3]
layers = 3;
grid3 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,66,67},
        PlotStyle->{Blue}];
```




```
N[FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
         == 0, {a, 67}],20]
```

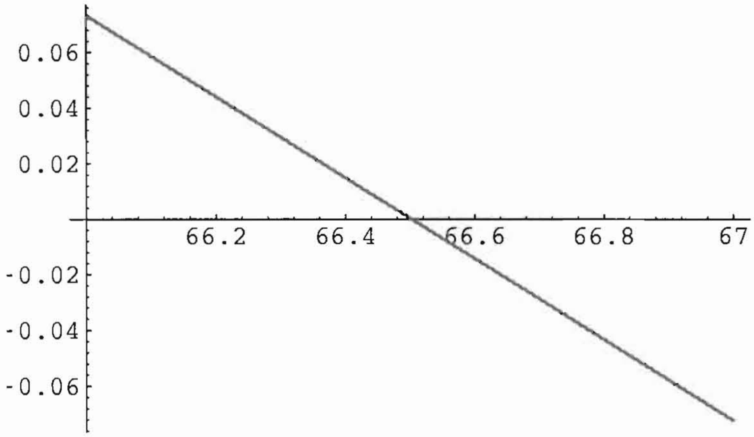
```
{a → 66.50222270248989}
```

```
Precision[%]
```

```
16
```

■ 7x7 Grid

```
Clear[layers,a,grid7]
layers = 7;
grid7 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,66,67},
        PlotStyle->{Green}];
```




```
N[FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
          == 0, {a, 67}],20]
```

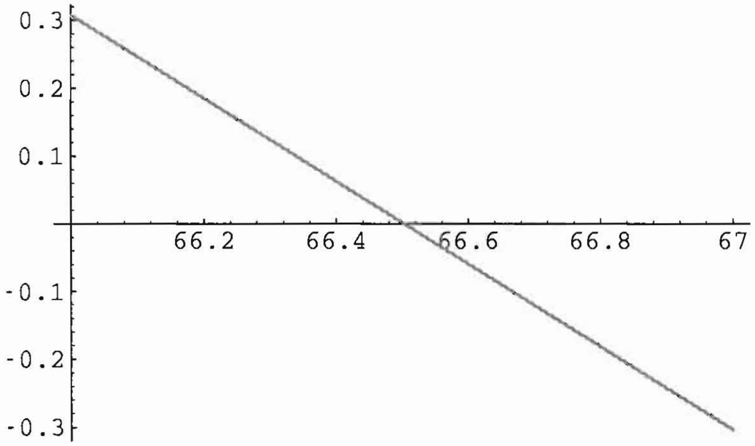
```
{a → 66.50222270248958}
```

```
Precision[%]
```

```
16
```

■ 15x15 Grid

```
Clear[layers,a,grid15]
layers = 15;
grid15 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,66,67},
        PlotStyle->{Cyan}];
```




```
N[FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}],20]
```

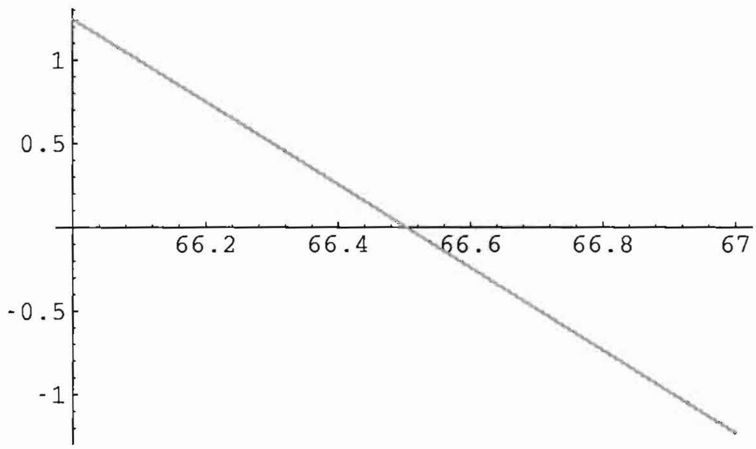
```
{a → 66.50222270248991}
```

```
Precision[%]
```

```
16
```

■ 31x31 Grid

```
Clear[layers,a,grid31]
layers = 31;
grid31 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,66,67},
        PlotStyle->{Banana}];
```




```
N[FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}],20]
```

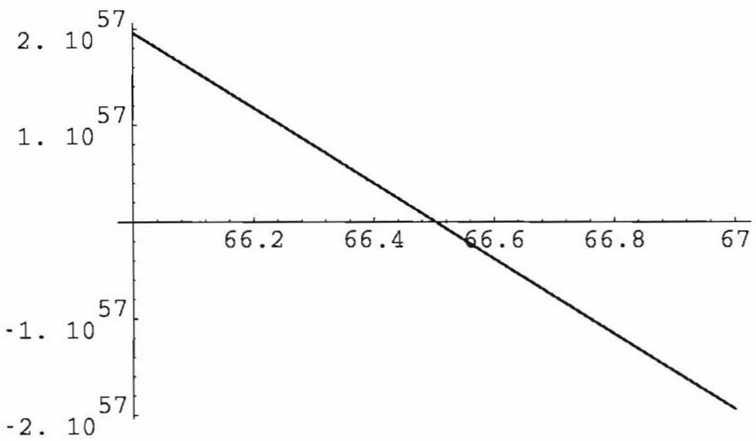
```
{a → 66.50222270248985}
```

```
Precision[%]
```

```
16
```

■ (2^100−1)x(2^100−1) Grid

```
Clear[layers,a,grid100]
layers = 2^100-1;
grid100 = Plot[
        rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
        {a,66,67},
        PlotStyle->{Red}];
```




```
N[FindRoot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers]
        == 0, {a, 67}],20]
```

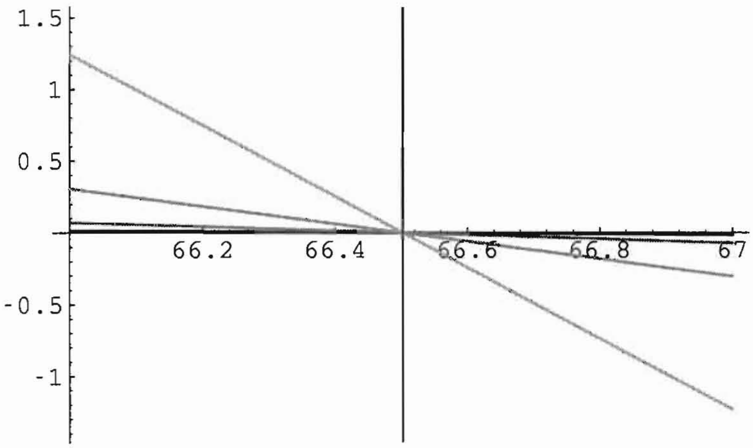
```
{a → 66.50222270248985}
```

```
Precision[%]
```

```
16
```

■ (Graph) Comparison

```
Show[grid3,grid7,grid15,grid31,grid100];
```



It appears that each grid has the same crossover point, even close up.

High Precision Comparison of "Powers of 2" Grids of Size (2^k – 1) for k = 2 to k = 100

Program to compare the crossover points found on other girds with number of points = power of 2. Creates a table of the resulting crossover points.

■ Initialization

```
(* set up notebook *)
Off[General::spell]
Needs["Graphics‘Colors‘"]

(* degree-to-radian conversion *)
Clear[d,DegRad];
DegRad[d_] := (Pi/180) d;

(* length of rhombus steiner tree *)
Clear[rhomtree];
rhomtree[a_] := 2*((-1/2 + Cos[a/2]^2)^2 + (3^(1/2)/2 +
                Sin[a]/2)^2)^(1/2);

(* length of triangle steiner tree *)
Clear[tritree];
tritree[a_] := Sin[(DegRad[180]-a)/2] +
              Sqrt[3] Cos[(DegRad[180]-a)/2];

(* number of rhombuses for grid *)
Clear[numrhom];
numrhom[layers_] := (4^(Log[layers+1]/Log[2]) - 1) / 3;

(* number of triangles for grid *)
numtri[layers_] :=
Module[{k,ntri,nrhom},
       If[Mod[layers,2] == 0,
          {k = layers/2; ntri = 2 k (k+1); nrhom = 0},
             (* even number of layers *)
          {k = (layers-1)/2; ntri = 2 ((k+1)^2-1); nrhom = 1}
             (* odd number of layers *)
       ];
       {ntri,nrhom}
];

(* length of rhombus grid tree *)
Clear[rhomgrid]
rhomgrid[a_,layers_] := numrhom[layers] rhomtree[a];

(* length of triangle grid tree *)
Clear[trigrid];
trigrid[a_,layers_] := numtri[layers][[1]] tritree[a] +
                       numtri[layers][[2]] rhomtree[a];
```

■ Grids

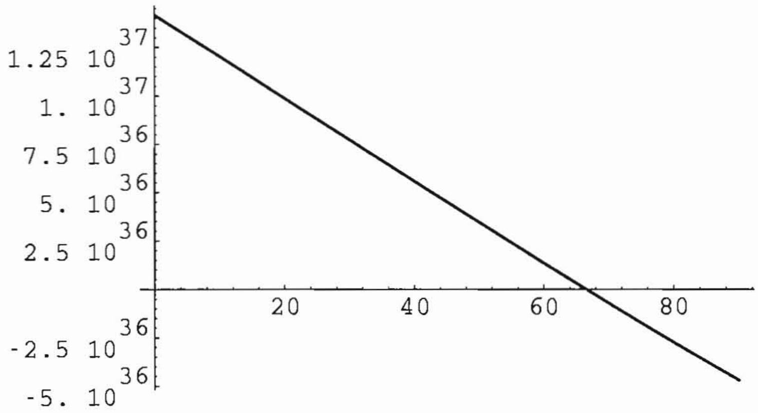
```
Clear[k]

(* set up table *)
Print["k     Calculated Crossover     Precision     Test"]

Do[
{ Clear[layers,a,result];
  layers = 2^k-1;
  (* calculate the crossover angle *)
  result = N[FindRoot[rhomgrid[DegRad[a],layers]-
             trigrid[DegRad[a],layers] == 0,
             {a, 66.502222}],20];
  (* print the result *)
  Print[k,"     ",result,"     ",Precision[result],"          ",
        N[rhomgrid[DegRad[result[[1,2]]],layers]
          -trigrid[DegRad[result[[1,2]]],layers]]
  ];
},
(* start k at 2 and take it through 100 *)
{k,2,100}
]
```

```
k     Calculated Crossover     Precision     Test
2     {a -> 66.50222270248985}     16         0.
                                                  -15
3     {a -> 66.50222270248958}     16         7.10543 10
                                                  -14
4     {a -> 66.50222270248985}     16         2.84217 10
5     {a -> 66.50222270248987}     16         0.
6     {a -> 66.50222270248942}     16         0.
                                                  -12
7     {a -> 66.50222270248938}     16         1.81899 10
8     {a -> 66.50222270248984}     16         0.
9     {a -> 66.50222270248982}     16         0.
                                                  -10
10     {a -> 66.50222270248992}     16         1.16415 10
11     {a -> 66.50222270248984}     16         0.
12     {a -> 66.50222270248897}     16         0.
13     {a -> 66.50222270248897}     16         0.
14     {a -> 66.50222270248895}     16         0.
                                                  -7
15     {a -> 66.50222270248985}     16         -1.19209 10
16     {a -> 66.50222270248985}     16         0.
17     {a -> 66.50222270248985}     16         0.
18     {a -> 66.50222270248985}     16         0.
19     {a -> 66.50222270248994}     16         0.
20     {a -> 66.50222270248985}     16         0.
21     {a -> 66.50222270248985}     16         -0.000488281
22     {a -> 66.50222270248985}     16         0.
23     {a -> 66.50222270248985}     16         0.
24     {a -> 66.50222270248802}     16         0.
25     {a -> 66.50222270248994}     16         0.
26     {a -> 66.50222270248803}     16         0.5
27     {a -> 66.50222270248994}     16         0.
28     {a -> 66.50222270248803}     16         8.
29     {a -> 66.50222270248985}     16         0.
30     {a -> 66.50222270248985}     16         0.
31     {a -> 66.50222270248985}     16         0.
32     {a -> 66.50222270248985}     16         0.
33     {a -> 66.50222270248985}     16         0.
34     {a -> 66.50222270248985}     16         0.
35     {a -> 66.50222270248985}     16         0.
36     {a -> 66.50222270248985}     16         0.
37     {a -> 66.50222270248985}     16         0.
38     {a -> 66.50222270248985}     16         0.
39     {a -> 66.50222270248985}     16         0.
40     {a -> 66.50222270248985}     16         0.
41     {a -> 66.50222270248985}     16         0.
42     {a -> 66.50222270248985}     16         0.
43     {a -> 66.50222270248985}     16         0.
44     {a -> 66.50222270248985}     16         0.
45     {a -> 66.50222270248985}     16         0.
46     {a -> 66.50222270248985}     16         0.
47     {a -> 66.50222270248985}     16         0.
                                                  12
48     {a -> 66.50222270248629}     16         -8.79609 10
49     {a -> 66.50222270248985}     16         0.
50     {a -> 66.50222270248985}     16         0.
51     {a -> 66.50222270248985}     16         0.
                                                  15
```

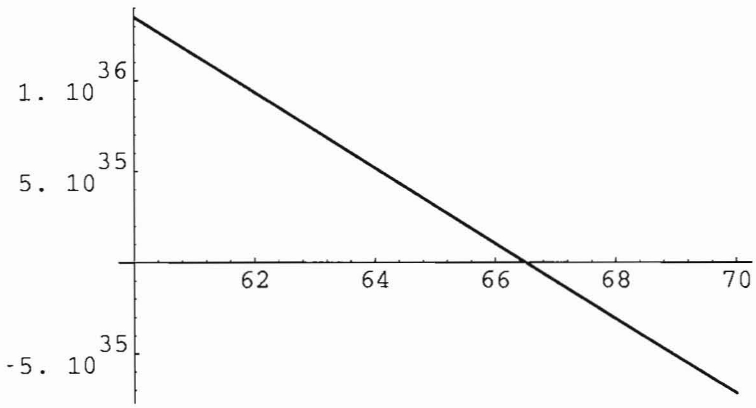
```
52      {a -> 66.50222270248629}     16          -2.2518 10
53      {a -> 66.50222270248985}     16          0.
54      {a -> 66.50222270248985}     16          0.
                                                           17
55      {a -> 66.5022227024935}     16         -1.44115 10
                                                            17
56      {a -> 66.50222270248629}     16          -5.76461 10
57      {a -> 66.50222270248985}     16          0.
58      {a -> 66.50222270248985}     16          0.
                                                           19
59      {a -> 66.5022227024935}     16         -3.68935 10
60      {a -> 66.50222270248985}     16          0.
61      {a -> 66.50222270248985}     16          0.
62      {a -> 66.50222270248985}     16          0.
                                                            21
63      {a -> 66.50222270248629}     16          -9.44473 10
64      {a -> 66.50222270248985}     16          0.
65      {a -> 66.50222270248985}     16          0.
66      {a -> 66.50222270248985}     16          0.
67      {a -> 66.50222270248985}     16          0.
68      {a -> 66.50222270248985}     16          0.
69      {a -> 66.50222270248985}     16          0.
70      {a -> 66.50222270248985}     16          0.
71      {a -> 66.50222270248985}     16          0.
72      {a -> 66.50222270248985}     16          0.
73      {a -> 66.50222270248985}     16          0.
74      {a -> 66.50222270248985}     16          0.
75      {a -> 66.50222270248985}     16          0.
76      {a -> 66.50222270248985}     16          0.
77      {a -> 66.50222270248985}     16          0.
78      {a -> 66.50222270248985}     16          0.
79      {a -> 66.50222270248985}     16          0.
80      {a -> 66.50222270248985}     16          0.
81      {a -> 66.50222270248985}     16          0.
82      {a -> 66.50222270248985}     16          0.
83      {a -> 66.50222270248985}     16          0.
84      {a -> 66.50222270248985}     16          0.
85      {a -> 66.50222270248985}     16          0.
86      {a -> 66.50222270248985}     16          0.
87      {a -> 66.50222270248985}     16          0.
88      {a -> 66.50222270248985}     16          0.
89      {a -> 66.50222270248985}     16          0.
90      {a -> 66.50222270248985}     16          0.
91      {a -> 66.50222270248985}     16          0.
92      {a -> 66.50222270248985}     16          0.
93      {a -> 66.50222270248985}     16          0.
94      {a -> 66.50222270248985}     16          0.
95      {a -> 66.50222270249717}     16          0.
96      {a -> 66.50222270248263}     16          0.
97      {a -> 66.50222270248985}     16          0.
98      {a -> 66.50222270248985}     16          0.
99      {a -> 66.50222270248985}     16          0.
100      {a -> 66.50222270248985}     16          0.

FindRoot::frmp:
   Machine precision is insufficient to achieve the accuracy
         -6
    1. 10   .

FindRoot::frmp:
   Machine precision is insufficient to achieve the accuracy
         -6
    1. 10   .

FindRoot::frmp:
   Machine precision is insufficient to achieve the accuracy
         -6
    1. 10   .

General::stop:
   Further output of FindRoot::frmp
     will be suppressed during this calculation.
```

Check grid sizes where difference appears to be "off":

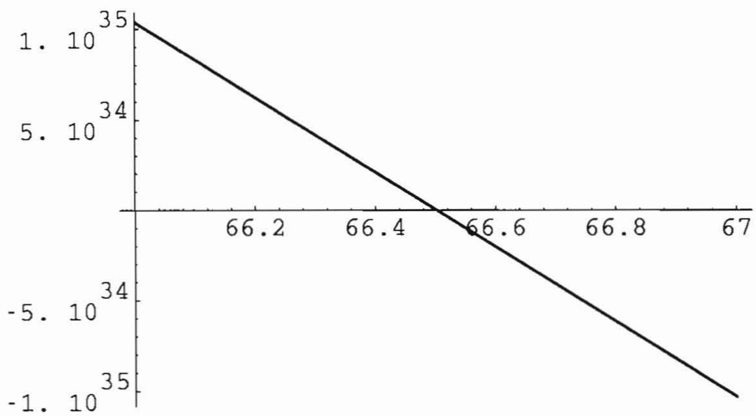
```
Clear[layers,k,result]
k = 63;
layers = 2^k-1;
Plot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
      {a,0,90}];
```




```
Plot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
     {a,60,70}];
```




```
Plot[rhomgrid[DegRad[a],layers]-trigrid[DegRad[a],layers],
     {a,66,67}];
```



```
result = N[FindRoot[rhomgrid[DegRad[a],layers]-
            trigrid[DegRad[a],layers] == 0,
            {a, 66.5}]]
```

{a → 66.50222270248622}

```
N[rhomgrid[DegRad[result[[1,2]]],layers]-
  trigrid[DegRad[result[[1,2]]],layers]]
```

$-(9.44473296573929\ 10^{21})$

The crossover values appear to be accurate through the 12th decimal place -- roundoff error or a slowly changing crossover angle?

Looking for Exact Value for Crossover Angle

Notebook to attempt to solve for an exact value for the crossover angle using the equation rhomtree[a]/3 – tritree[a]/2 = 0

■ Solving for Exact Value of Crossover Angle

By simplifying by hand, we have reduced the equation
rhomtree[a]/3 – tritree[a]/2 == 0
down to
(9/4) Cos[a] == (1/2) + (Sqrt[3]/4) Sin[a] . (eq1)
Using this equation, we attempt to have Mathematica solve for the angle a:

```
Clear[a]
Solve[(9/4) Cos[a] == (1/2) + (Sqrt[3]/4) Sin[a], a]
```

```
Solve::tdep:
   The equations appear to involve transcendental functions of
     the variables in an essentially non-algebraic way.
```

$$\text{Solve}\left[\frac{9\,\text{Cos}[a]}{4} == \frac{1}{2} + \frac{1}{4}\sqrt{3}\,\text{Sin}[a],\ a\right]$$

Mathematica didn't like that. We then move onto our second tactic: substituting x = Sin[a] into the equation (making Cos[x] = Sqrt[1–x^2] –– note that this is legal because we know that a is the small angle on our rhombus and thus 0 <= a <= 90 degrees always). (eq1) then becomes:
(9/4) Sqrt[1–x^2] == (1/2) + (Sqrt[3]/4) x . (eq2)
We will know be solving for Sin[a] rather than a:

```
Clear[x, sol]
sol = Solve[(9/4) Sqrt[1-x^2] == (1/2) + (Sqrt[3]/4) x, x]
```

$$\left\{\left\{x \to \frac{1}{168}\left(-4\sqrt{3} - 72\sqrt{5}\right)\right\},\ \left\{x \to \frac{1}{168}\left(-4\sqrt{3} + 72\sqrt{5}\right)\right\}\right\}$$

Mathematica returns two results; we want the positive square root result because 0 <= a <= 90 degrees => 0 <= Sin[a] <= 1 .
So,

```
x = sol[[2,1,2]]
```

$$\frac{1}{168}\left(-4\sqrt{3} + 72\sqrt{5}\right)$$

```
N[x]
```

```
0.917075542557794
```

```
N[Sqrt[3]/2]
```

```
0.866025403784439
```

Notice that our solution is slightly larger than Sqrt[3]/2, as we expected --
since the numeric approximation of our angle is around 66.50222...,
Sin[66.50222... degrees] should be slightly larger than Sin[60 degrees] = Sqrt[3]/2. Therefore

Sin[crossover] = (72 Sqrt[5] – 4 Sqrt[3])/168

=>

crossover = ArcSin[(72 Sqrt[5] – 4 Sqrt[3])/168]:

```
Clear[crossover]
crossover = ArcSin[x]
```

$$\text{ArcSin}\left[\frac{1}{168}\left(-4\sqrt{3}+72\sqrt{5}\right)\right]$$

```
N[crossover 180/Pi,20]   (* convert from radians to degrees *)
```

```
66.50222270248991235
```

```
Accuracy[%]
```

```
17
```

The result really does give the numeric value we were getting for our crossover angle.

■ Rewriting Crossover Angle (Exact Computation)

Consider x = Sin[crossover]:

```
x
```

$$\frac{1}{168}\left(-4\sqrt{3}+72\sqrt{5}\right)$$

Notice that we can rewrite x as follows:

x = –4 Sqrt[3]/168 + 72 Sqrt[5]/168
= (–4/84) Sqrt[3]/2 + (72 Sqrt[5]/84) (1/2)
= (–1/21) Sin[60 Degree] + (6 Sqrt[5]/7) Cos[60 Degree]

or

x = –4 Sqrt[3]/168 + 72 Sqrt[5]/168
= (–4/84) Sqrt[3]/2 + (72 Sqrt[5]/(84 Sqrt[3])) (Sqrt[3]/2)
= (–1/21) Sin[60 Degree] + (6 Sqrt[5]/(7 Sqrt[3])) Sin[60 Degree]
= Sin[60 Degree] (6 Sqrt[5]/(7 Sqrt[3]) – 1/21)

APPENDIX C

LADDERS PROGRAMS

This appendix contains the *Mathematica* programs that draw and calculate the length of Steiner trees over both square and generalized 1×4 ladders.

The basic idea behind these programs is to solve for the location of the Steiner points based on the restrictions imposed upon the Steiner tree by the ladder structure (given in [3]) and by basic Steiner point properties. The the resulting Steiner tree over the 1×4 square ladder is shown in Figure C.1. The a_i's and b_i's represent network points and the s_i's represent Steiner points. The m_i's represent the slopes of the lines. From [3],

$$m_0 = ((\ell + 1)(2 + \sqrt{3}) - 2)^{-1}.$$

This, using the fact that opposite angles are congruent and the fact that all Steiner points meet at angles of 120°, gives the rest of the slopes:

$$m_0 = m_6 = m_{10} = m_{14},$$

$$m_1 = m_4 = m_8 = m_9 = m_{12} = m_{16} = -\tan(60^\circ - \arctan(m_0)),$$

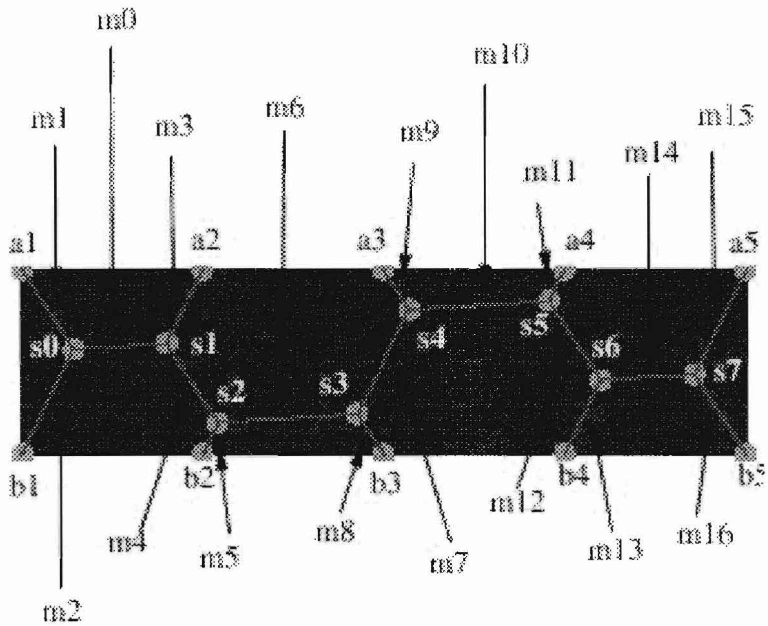


Figure C.1 The strucure of the Steiner tree over a 1×4 ladder.

and

$$m_2 = m_3 = m_5 = m_7 = m_{11} = m_{13} = m_{15} = \tan(60^\circ + \arctan(m_0)).$$

These slopes enable us to set up a series of equations to solve for the location of each Steiner point, as done in the square ladder program. The rhombus ladder program uses a similar set up, but changes the definition of m_0 to return the (experimentally) best generalized ladder.

Rhombus Ladders

Attempt 2.1: n = 5 case for 90 Degrees

This notebook is a verification of the process, using smangle = 90 Degrees to check against the ladder article's length.

Initialization Cells

```
(* setting up notebook *)
  Off[General::spell]
  Needs["Graphics'Colors'"]

(* distance function *)
  Clear[Distance,x1,y1,x2,y2];
  Distance[{x1_, y1_}, {x2_, y2_}] =
            Sqrt[(x1-x2)^2 + (y1-y2)^2];

(* degree to radian conversion -- since Mathematica's
   default is radians *)
  Clear[d,DegRad];
  DegRad[d_] = (Pi/180) d;
```

Beginning Notes

The ladder that we will be building a Steiner tree for the n = 5 case (putting four rhombuses together). For this notebook, we will be modifying the notation given in Chung and Graham's *Steiner Trees for Ladders* for the ladder points: the lower points of the ladder, labeled as *b*'s, will lie on the y = 0 line, while the upper points, labeled as *a*'s, will lie on the y = 2Sin[smangle] line. Steiner points will be labeled as *s*'s. The slopes of lines will be labeled as *m*'s.

Problem Setup for smanlge = 90 Degrees

```
Clear[smangle,angle,n,a1,a2,a3,a4,a5,b1,b2,b3,b4,b5]
smangle = 90;
angle = DegRad[smangle];

n = 5;

a1 = {2 Cos[angle], 2 Sin[angle]};
a2 = {2 + 2 Cos[angle], 2 Sin[angle]};
a3 = {4 + 2 Cos[angle], 2 Sin[angle]};
a4 = {6 + 2 Cos[angle], 2 Sin[angle]};
a5 = {8 + 2 Cos[angle], 2 Sin[angle]};

b1 = {0,0};
b2 = {2,0};
b3 = {4,0};
b4 = {6,0};
b5 = {8,0};

Clear[s0,s1,s2,s3,s4,s5,s6,s7]
```

Slopes

```
Clear[m0,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,
      m15,m16]
m0 = (n (2+Sqrt[3]) - 2)^(-1);
m6 = m0;
m10 = m0;
m14 = m0;

m1 = -Tan[DegRad[60] - ArcTan[m0]];
m4 = m1;
m8 = m1;
m9 = m1;
m12 = m1;
m16 = m1;

m2 = Tan[DegRad[60] + ArcTan[m0]];
m3 = m2;
m5 = m2;
m7 = m2;
m11 = m2;
m13 = m2;
m15 = m2;
```

Define Steiner Points

```
Clear[eq1,eq2,x,y]
eq1[x_] = a1[[2]] + m1 (x-a1[[1]]);
eq2[x_] = b1[[2]] + m2 (x-b1[[1]]);
x = NSolve[eq1[x] == eq2[x],x][[1,1,2]];
y = eq1[x];
s0 = {x,y}
```

$$\left\{0.5690598923241498,\ 2 - 0.5690598923241498\,\mathrm{Tan}\left[\frac{\pi}{3} - \mathrm{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\right\}$$

```
Clear[eq3,eq4,x,y]
eq3[x_] = s0[[2]] + m0 (x-s0[[1]]);
eq4[x_] = a2[[2]] + m3 (x-a2[[1]]);
x = NSolve[eq3[x] == eq4[x],x][[1,1,2]];
y = eq3[x];
s1 = {x,y}
```

$$\left\{1.6,\ 2 + \frac{1.03094010767585}{-2+5\left(2+\sqrt{3}\right)} - 0.5690598923241498\,\mathrm{Tan}\left[\frac{\pi}{3} - \mathrm{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\right\}$$

```
Clear[eq5,eq6,x,y]
eq5[x_] = s1[[2]] + m4 (x-s1[[1]]);
eq6[x_] = b2[[2]] + m5 (x-b2[[1]]);
x = NSolve[eq5[x] == eq6[x],x][[1,1,2]];
y = eq5[x];
s2 = {x,y}
```

$$\{2.169059892324149,\ 2+\frac{1.03094010767585}{-2+5\left(2+\sqrt{3}\right)}-1.138119784648299\,\text{Tan}\left[\frac{\pi}{3}-\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\}$$

```
Clear[eq7,eq8,x,y]
eq7[x_] = s2[[2]] + m6 (x-s2[[1]]);
eq8[x_] = b3[[2]] + m8 (x-b3[[1]]);
x = NSolve[eq7[x] == eq8[x],x][[1,1,2]];
y = eq7[x];
s3 = {x,y}
```

$$\{3.715470053837925,\ 2+\frac{2.577350269189625}{-2+5\left(2+\sqrt{3}\right)}-1.138119784648299\,\text{Tan}\left[\frac{\pi}{3}-\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\}$$

```
Clear[eq9,eq10,x,y]
eq9[x_] = s3[[2]] + m7 (x-s3[[1]]);
eq10[x_] = a3[[2]] + m9 (x-a3[[1]]);
x = NSolve[eq9[x] == eq10[x],x][[1,1,2]];
y = eq9[x];
s4 = {x,y}
```

$$\{4.284529946162074,\ 2+\frac{2.577350269189625}{-2+5\left(2+\sqrt{3}\right)}-1.138119784648299\,\text{Tan}\left[\frac{\pi}{3}-\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]+0.5690598923241495\,\text{Tan}\left[\frac{\pi}{3}+\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\}$$

```
Clear[eq11,eq12,x,y]
eq11[x_] = s4[[2]] + m10 (x-s4[[1]]);
eq12[x_] = a4[[2]] + m11 (x-a4[[1]]);
x = NSolve[eq11[x] == eq12[x],x][[1,1,2]];
y = N[eq11[x]];
s5 = {x,y}
```

{5.830940107675851, 1.6618802153517}

```
Clear[eq13,eq14,x,y]
eq13[x_] = s5[[2]] + m12 (x-s5[[1]]);
eq14[x_] = b4[[2]] + m13 (x-b4[[1]]);
x = NSolve[eq13[x] == eq14[x],x][[1,1,2]];
y = eq13[x];
s6 = {x,y}
```

$$\{6.4,\ 1.6618802153517-0.569059892324149\,\text{Tan}\left[\frac{\pi}{3}-\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\}$$

```
Clear[eq15,eq16,x,y]
eq15[x_] = a5[[2]] + m15 (x-a5[[1]]);
eq16[x_] = b5[[2]] + m16 (x-b5[[1]]);
x = NSolve[eq15[x] == eq16[x],x][[1,1,2]];
y = eq15[x];
s7 = {x,y}
```

$$\{7.43094010767585,\ 2-0.5690598923241508\,\text{Tan}\left[\frac{\pi}{3}+\text{ArcTan}\left[\frac{1}{-2+5\left(2+\sqrt{3}\right)}\right]\right]\}$$

Length of Steiner Tree Over Ladder

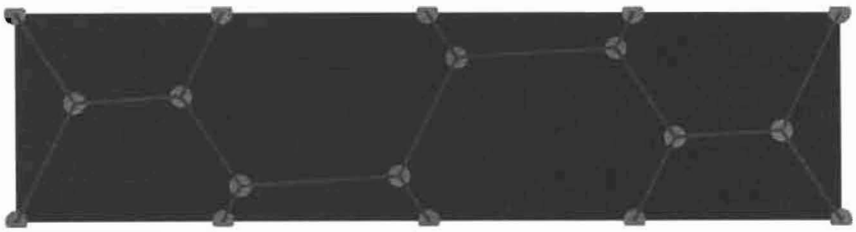
```
N[Distance[a1,s0] + Distance[b1,s0] + Distance[s0,s1] +
  Distance[s1,a2] + Distance[s1,s2] + Distance[b2,s2] +
  Distance[s2,s3] + Distance[s3,s4] + Distance[s3,b3] +
  Distance[s4,a3] + Distance[s4,s5] + Distance[s5,a4] +
  Distance[s5,s6] + Distance[s6,b4] + Distance[s6,s7] +
  Distance[a5,s7] + Distance[b5,s7],15]
```

```
16.69023860241399
```

Graphics

Here’s a picture of our result:

```
Clear[squares]
squares =
{Graphics[{Blue,Polygon[{a1,b1,b2,a2}]}],
 Graphics[{Blue,Polygon[{a2,b2,b3,a3}]}],
 Graphics[{Blue,Polygon[{a3,b3,b4,a4}]}],
 Graphics[{Blue,Polygon[{a4,b4,b5,a5}]}]};

Clear[steinerpts]
steinerpts =
{Graphics[{Green,PointSize[0.025],Point[s0]}],
 Graphics[{Green,PointSize[0.025],Point[s1]}],
 Graphics[{Green,PointSize[0.025],Point[s2]}],
 Graphics[{Green,PointSize[0.025],Point[s3]}],
 Graphics[{Green,PointSize[0.025],Point[s4]}],
 Graphics[{Green,PointSize[0.025],Point[s5]}],
 Graphics[{Green,PointSize[0.025],Point[s6]}],
 Graphics[{Green,PointSize[0.025],Point[s7]}]};

Clear[ladderpts]
ladderpts =
{Graphics[{Green,PointSize[0.025],Point[a1]}],
 Graphics[{Green,PointSize[0.025],Point[a2]}],
 Graphics[{Green,PointSize[0.025],Point[a3]}],
 Graphics[{Green,PointSize[0.025],Point[a4]}],
 Graphics[{Green,PointSize[0.025],Point[a5]}],
 Graphics[{Green,PointSize[0.025],Point[b1]}],
 Graphics[{Green,PointSize[0.025],Point[b2]}],
 Graphics[{Green,PointSize[0.025],Point[b3]}],
 Graphics[{Green,PointSize[0.025],Point[b4]}],
 Graphics[{Green,PointSize[0.025],Point[b5]}]};

Clear[lines]
lines = {Graphics[{Red,Line[{a1,s0}]}],
         Graphics[{Red,Line[{b1,s0}]}],
         Graphics[{Red,Line[{s0,s1}]}],
         Graphics[{Red,Line[{s1,a2}]}],
         Graphics[{Red,Line[{s1,s2}]}],
         Graphics[{Red,Line[{b2,s2}]}],
         Graphics[{Red,Line[{s2,s3}]}],
         Graphics[{Red,Line[{s3,s4}]}],
         Graphics[{Red,Line[{b3,s3}]}],
         Graphics[{Red,Line[{a3,s4}]}],
         Graphics[{Red,Line[{s4,s5}]}],
         Graphics[{Red,Line[{s5,a4}]}],
         Graphics[{Red,Line[{s5,s6}]}],
         Graphics[{Red,Line[{s6,b4}]}],
         Graphics[{Red,Line[{s6,s7}]}],
         Graphics[{Red,Line[{a5,s7}]}],
         Graphics[{Red,Line[{b5,s7}]}]};

Show[squares,steinerpts,ladderpts,lines,
     AspectRatio->Automatic];
```



Rhombus Ladders

Attempt 2.2: Perfect Diamond with n = 5
Length of rhombus side = 1

Initialization Cells

```
(* setting up notebook *)
  Off[General::spell]
  Needs["Graphics'Colors'"]

(* distance function *)
  Clear[Distance,x1,y1,x2,y2];
  Distance[{x1_, y1_}, {x2_, y2_}] =
          Sqrt[(x1-x2)^2 + (y1-y2)^2];

(* degree to radian conversion -- since Mathematica's
   default is radians *)
  Clear[d,DegRad];
  DegRad[d_] = (Pi/180) d;
```

Beginning Notes

The ladder that we will be building a Steiner tree for the n = 5 case (putting four rhombuses together). For this notebook, we will be modifying the notation given in Chung and Graham's *Steiner Trees for Ladders* for the ladder points: the lower points of the ladder, labeled as *b*'s, will lie on the y = 0 line, while the upper points, labeled as *a*'s, will lie on the y = Sin[smangle] line. Steiner points will be labeled as *s*'s. The slopes of lines will be labeled as *m*'s.

Problem Setup for smanlge = 60 Degrees

```
Clear[smangle,angle,n,a1,a2,a3,a4,a5,b1,b2,b3,b4,b5]
smangle = 60;

angle = DegRad[smangle];

n = 5;

a1 = {Cos[angle], Sin[angle]};
a2 = {1 + Cos[angle], Sin[angle]};
a3 = {2 + Cos[angle], Sin[angle]};
a4 = {3 + Cos[angle], Sin[angle]};
a5 = {4 + Cos[angle], Sin[angle]};

b1 = {0,0};
b2 = {1,0};
b3 = {2,0};
b4 = {3,0};
b5 = {4,0};

Clear[s0,s1,s2,s3,s4,s5,s6,s7]
```

Slopes

```
Clear[m0,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,
      m15,m16]
m0 = -0.21;

m6 = m0;
m10 = m0;
m14 = m0;

m1 = -Tan[DegRad[60] - ArcTan[m0]];
m4 = m1;
m8 = m1;
m9 = m1;
m12 = m1;
m16 = m1;

m2 = Tan[DegRad[60] + ArcTan[m0]];
m3 = m2;
m5 = m2;
m7 = m2;
m11 = m2;
m13 = m2;
m15 = m2;
```

Define Steiner Points

```
Clear[eq1,eq2,x,y]
eq1[x_] = a1[[2]] + m1 (x-a1[[1]]);
eq2[x_] = b1[[2]] + m2 (x-b1[[1]]);
x = NSolve[eq1[x] == eq2[x],x][[1,1,2]];
y = eq1[x];
s0 = {x,y}
```

$$\left\{0.5738852183984498,\ \frac{\sqrt{3}}{2} - 0.07388521839844975\,\mathrm{Tan}\left[0.206992194219821 + \frac{\pi}{3}\right]\right\}$$

```
Clear[eq3,eq4,x,y]
eq3[x_] = s0[[2]] + m0 (x-s0[[1]]);
eq4[x_] = a2[[2]] + m3 (x-a2[[1]]);
x = NSolve[eq3[x] == eq4[x],x][[1,1,2]];
y = eq3[x];
s1 = {x,y}
```

$$\left\{1.183280228848154,\ -0.1279729521944378 + \frac{\sqrt{3}}{2} - 0.07388521839844975\,\mathrm{Tan}\left[0.206992194219821 + \frac{\pi}{3}\right]\right\}$$

```
Clear[eq5,eq6,x,y]
eq5[x_] = s1[[2]] + m4 (x-s1[[1]]);
eq6[x_] = b2[[2]] + m5 (x-b2[[1]]);
x = NSolve[eq5[x] == eq6[x],x][[1,1,2]];
y = eq5[x];
s2 = {x,y}
```

$\{1.257165447246603,$
$-0.1279729521944378+\frac{\sqrt{3}}{2}-0.1477704367968993\,\mathrm{Tan}[0.206992194219821+\frac{\pi}{3}]\}$

```
Clear[eq7,eq8,x,y]
eq7[x_] = s2[[2]] + m6 (x-s2[[1]]);
eq8[x_] = b3[[2]] + m8 (x-b3[[1]]);
x = NSolve[eq7[x] == eq8[x],x][[1,1,2]];
y = eq7[x];
s3 = {x,y}
```

$\{1.953900759169722,$
$-0.2742873676982928+\frac{\sqrt{3}}{2}-0.1477704367968993\,\mathrm{Tan}[0.206992194219821+\frac{\pi}{3}]\}$

```
Clear[eq9,eq10,x,y]
eq9[x_] = s3[[2]] + m7 (x-s3[[1]]);
eq10[x_] = a3[[2]] + m9 (x-a3[[1]]);
x = NSolve[eq9[x] == eq10[x],x][[1,1,2]];
y = eq9[x];
s4 = {x,y}
```

$\{2.527785977568171,$
$-0.2742873676982928+\frac{\sqrt{3}}{2}-0.5738852183984495\,\mathrm{Tan}[0.206992194219821-\frac{\pi}{3}]-$
$0.1477704367968993\,\mathrm{Tan}[0.206992194219821+\frac{\pi}{3}]\}$

```
Clear[eq11,eq12,x,y]
eq11[x_] = s4[[2]] + m10 (x-s4[[1]]);
eq12[x_] = a4[[2]] + m11 (x-a4[[1]]);
x = NSolve[eq11[x] == eq12[x],x][[1,1,2]];
y = N[eq11[x]];
s5 = {x,y}
```

{3.282085790518834, 0.6228127915599226}

```
Clear[eq13,eq14,x,y]
eq13[x_] = s5[[2]] + m12 (x-s5[[1]]);
eq14[x_] = b4[[2]] + m13 (x-b4[[1]]);
x = NSolve[eq13[x] == eq14[x],x][[1,1,2]];
y = eq13[x];
s6 = {x,y}
```

$\{3.355971008917284,\ 0.6228127915599226-0.07388521839844975\,\mathrm{Tan}[0.206992194219821+\frac{\pi}{3}]\}$

```
Clear[eq15,eq16,x,y]
eq15[x_] = a5[[2]] + m15 (x-a5[[1]]);
eq16[x_] = b5[[2]] + m16 (x-b5[[1]]);
x = NSolve[eq15[x] == eq16[x],x][[1,1,2]];
y = eq15[x];
s7 = {x,y}
```

$\{3.926114781601551,\ \frac{\sqrt{3}}{2}+0.5738852183984493\,\mathrm{Tan}[0.206992194219821-\frac{\pi}{3}]\}$

Length of Steiner Tree Over Ladder

```
N[Distance[a1,s0] + Distance[b1,s0] + Distance[s0,s1] +
  Distance[s1,a2] + Distance[s1,s2] + Distance[b2,s2] +
  Distance[s2,s3] + Distance[s3,s4] + Distance[s3,b3] +
  Distance[s4,a3] + Distance[s4,s5] + Distance[s5,a4] +
  Distance[s5,s6] + Distance[s6,b4] + Distance[s6,s7] +
  Distance[a5,s7] + Distance[b5,s7],15]
```

```
8.18737713007286
```

Graphics

Here’s a picture of our result:

```
Clear[squares]
squares =
{Graphics[{Blue,Polygon[{a1,b1,b2,a2}]}],
 Graphics[{Blue,Polygon[{a2,b2,b3,a3}]}],
 Graphics[{Blue,Polygon[{a3,b3,b4,a4}]}],
 Graphics[{Blue,Polygon[{a4,b4,b5,a5}]}]};

Clear[steinerpts]
steinerpts =
{Graphics[{Green,PointSize[0.025],Point[s0]}],
 Graphics[{Green,PointSize[0.025],Point[s1]}],
 Graphics[{Green,PointSize[0.025],Point[s2]}],
 Graphics[{Green,PointSize[0.025],Point[s3]}],
 Graphics[{Green,PointSize[0.025],Point[s4]}],
 Graphics[{Green,PointSize[0.025],Point[s5]}],
 Graphics[{Green,PointSize[0.025],Point[s6]}],
 Graphics[{Green,PointSize[0.025],Point[s7]}]};

Clear[ladderpts]
ladderpts =
{Graphics[{Green,PointSize[0.025],Point[a1]}],
 Graphics[{Green,PointSize[0.025],Point[a2]}],
 Graphics[{Green,PointSize[0.025],Point[a3]}],
 Graphics[{Green,PointSize[0.025],Point[a4]}],
 Graphics[{Green,PointSize[0.025],Point[a5]}],
 Graphics[{Green,PointSize[0.025],Point[b1]}],
 Graphics[{Green,PointSize[0.025],Point[b2]}],
 Graphics[{Green,PointSize[0.025],Point[b3]}],
 Graphics[{Green,PointSize[0.025],Point[b4]}],
 Graphics[{Green,PointSize[0.025],Point[b5]}]};

Clear[lines]
lines = {Graphics[{Red,Line[{a1,s0}]}],
         Graphics[{Red,Line[{b1,s0}]}],
         Graphics[{Red,Line[{s0,s1}]}],
         Graphics[{Red,Line[{s1,a2}]}],
         Graphics[{Red,Line[{s1,s2}]}],
         Graphics[{Red,Line[{b2,s2}]}],
         Graphics[{Red,Line[{s2,s3}]}],
         Graphics[{Red,Line[{s3,s4}]}],
         Graphics[{Red,Line[{b3,s3}]}],
         Graphics[{Red,Line[{a3,s4}]}],
         Graphics[{Red,Line[{s4,s5}]}],
         Graphics[{Red,Line[{s5,a4}]}],
         Graphics[{Red,Line[{s5,s6}]}],
         Graphics[{Red,Line[{s6,b4}]}],
         Graphics[{Red,Line[{s6,s7}]}],
         Graphics[{Red,Line[{a5,s7}]}],
         Graphics[{Red,Line[{b5,s7}]}]};
```

```
Show[squares,steinerpts,ladderpts,lines,
     AspectRatio->Automatic];
```