A Stochastic Optimization Algorithm for Steiner Minimal Trees

Frederick C. Harris, Jr.
fredh@cs.clemson.edu

Department of Computer Science
Clemson University
Clemson, South Carolina 29634-1906

Abstract

The Optimization problem is simply stated as follows: Given a set of N cities, construct a connected network which has minimum length. The problem is simple enough, but the catch is that you are allowed to add junctions in your network. Therefore the problem becomes how many extra junctions should be added, and where should they be placed so as to minimize the overall network length.

This intriguing optimization problem is also known as the Steiner Minimal Tree Problem, where the junctions that are added to the network are called Steiner Points. A Simulated Annealing approach is proposed for this NP-Hard problem, and some very exciting results from it are presented.

1 The Problem

Minimizing a network's length is one of the oldest optimization problems in mathematics and, consequently, it has been worked on by many of the leading mathematicians in history. In the mid-seventeenth century a simple problem was posed: Find the point P that minimizes the sum of the distances from P to each of three given points in the plane. Solutions to this problem were derived independently by Fermat, Torricelli and Cavalieri. They all deduced that either P is inside the triangle formed by the given points and that the angles at P formed by the lines joining P to the three points are all $120°$, or P is one of the three vertices and the angle at P formed by the lines joining P to the other two points is greater than or equal to $120°$.

In the nineteenth century a mathematician at the University of Berlin, named Jakob Steiner, studied this problem and generalized it to include an arbitrarily large set of points in the plane. This generalization created a star when P was connected to all the given points in the plane, and is a geometric approach to the 2-dimensional center of mass problem.

In 1934 Kössler and Jarník generalized the network minimization problem even further [21]: Given n points in the plane find the shortest possible connected network containing these points. This generalized problem, however, did not become popular until the book, *What is Mathematics*, by Courant and Robbins [7], appeared in 1941. Courant and Robbins linked the name Steiner with this form of the problem proposed by Kössler and Jarník, and it became known as the Steiner Minimal Tree problem. The general solution to this problem allows multiple points to be added, each of which is called a Steiner Point, creating a tree instead of a star.

Much is known about the exact solution to the Steiner Minimal Tree problem. Those who wish to learn about some of the spin-off problems are invited to read the introductory article by Bern and Graham [1], the excellent survey paper on this problem by Hwang and Richards [19], or the recent volume in The Annals of Discrete Mathematics devoted completely to Steiner Tree problems [20]. Some of the basic pieces of information about the Steiner Minimal Tree problem that can be gleaned from these articles are: (i) the fact that all of the original n points will be of degree 1, 2, or 3, (ii) the Steiner Points are all of degree 3, (iii) any two edges meet at an angle of at least 120° in the Steiner Minimal Tree, and (iv) at most $n - 2$ Steiner Points will be added to the network.

2 The First Solution

A typical problem-solving approach is to begin with the simple cases and expand to a general solution. As we saw in Section 1, the trivial three point problem had already been solved in the 1600's, so all that remained was the work toward a general solution. As with many interesting problems this is harder than it appears on the surface.

The method proposed by the mathematicians of the mid-seventeenth century for the three point problem is illustrated in Figure 1. This method stated that in order to calculate the Steiner Point given points A, B, and C, you first construct an equilateral triangle (ACX) using the longest edge between two of the points (AC) such that the third (B) lies outside the triangle. A circle is circumscribed around the triangle, and a line is constructed from the third point (B) to the far vertex of the triangle (X). The location of the Steiner Point (P) is the intersection of this line (BX) with the circle.

For the next thirty years after Kössler and Jarník presented the general form of the SMT problem, the only algorithms that existed were heuristics. The heuristics were typically based upon the Minimum-Length Spanning Tree (MST), which is a tree that spans or connects all vertices whose sum of the edge lengths is as small as possible, and tried in various ways to join three vertices with a Steiner Point. In 1968 Gilbert and Pollak [16] linked the length of the SMT to the length of a MST. It was already known that the length of an MST is an upper bound for the length of an SMT, but their conjecture stated that the length of an SMT would never be any shorter than $\frac{\sqrt{3}}{2}$ times the length of an MST. This conjecture, was recently proved [8], and has led to the MST being the starting point for most of the heuristics that have been proposed in the last 20 years. This combination approach, starting with the

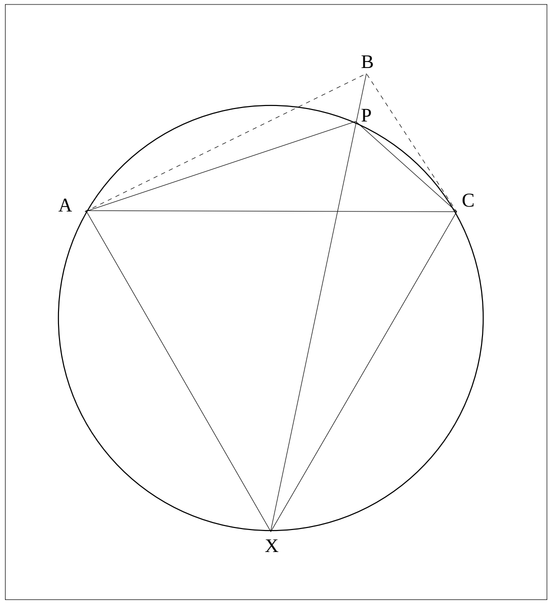


Figure 1: AP + CP = PX.

MST, will serve as the basis for the heuristic we present later in this paper.

In 1961 Melzak developed the first algorithm for calculating an SMT [23]. Melzak's Algorithm was geometric in nature and was based upon some simple extensions to Figure 1. The insight that Melzak offered was the fact that you can reduce an n point problem to a set of $n-1$ point problems. This reduction in size is accomplished by taking every pair of points, A and C in our example, calculating where the two possible points, X_1 and X_2, would be that form an equilateral triangle with them, and creating two smaller problems, one where X_1 replaces A and C, and the other where X_2 replaces A and C. Both Melzak and Cockayne pointed out however that some of these sub-problems are invalid. Melzak's algorithm can then be run on the two smaller problems. This recursion, based upon replacing two points with one point, finally terminates when you reduce the problem from three to two vertices. At this termination the length of the tree will be the length of the line segment connecting the final two points. This is due to the fact that $BP + AP + CP = BP + PX$. This is straightforward to prove using the law of cosines, for when P is on the circle, $\angle APX = \angle CPX = 60°$. This allows the calculation of the last Steiner Point (P) and allows you to back up the recursive call stack to calculate where each Steiner Point in that particular tree is located.

This reduction is important in the calculation of an SMT, but the algorithm still has exponential order, since it requires looking at every possible reduction of a pair of points to a single point. The recurrence relation for an n-point problem is stated quite simply in the following formula:

$$T(n) = 2 * \binom{n}{2} * T(n-1).$$

This yields what is obviously a non-polynomial time algorithm. In fact Garey, Graham, and Johnson [9] have shown that the Steiner Minimal Tree problem is NP-Hard (NP-Complete if the distances are rounded up to discrete values).

3 Current Results

In 1967, just a few years after Melzak's paper, Cockayne [2] clarified some of the details from Melzak's proof and proposed a new term, the Steiner Hull [3]. This allows the problem to be decomposed into subproblems based on regions contained in the hull, and the trees to be joined at the regions intersections. This decomposition, along with a clarified algorithm, proved to be the basis for the first computer program to calculate Steiner Minimal Trees, which appeared in [6], and which could compute an SMT for any placement of up to 7 vertices.

The biggest breakthrough after Cockayne's algorithm came from Winter in 1985 [28] who was able to establish some geometric results which showed that a possible sub-tree could not possibly exist in the Steiner Minimal Tree. This enabled one to eliminate (prune out) a great many of the reconstruction sequences required by Melzak's algorithm. Using this, Winter was able to compute Steiner Minimal Trees for 15 or fewer vertices. Winter's algorithm has been the basis for most of the enhancements in Steiner Minimal Tree algorithms since then.

In 1986 another major computational breakthrough was made. Cockayne and Hewgill [4] were able to calculate the Steiner Minimal Trees for up to 30 vertices 80% of the time. They achieved this through the use of what they termed an incompatibility matrix, which took the sub-trees left after Winter's pruning and determined whether trees i and j could appear together in the Steiner Minimal Tree.

Most of the rest of the exact current results belong to Cockayne and Hewgill. In 1992 they expanded upon their Incompatibility Matrix and developed better pruning techniques that have allowed them to calculate Steiner Minimal Trees for up to 100 vertices 80% of the time. Their paper describing these results [5] recently appeared in a Special Issue of *Algorithmica* which is totally devoted to the Steiner Problem.

4 The Proposed Heuristic

Background and Motivation

By exploring a structural similarity between *stochastic Petri nets* (see [25] and [24]) and *Hopfield neural nets* (see [17] and [18]), Geist was able to propose and take part in the development of a new computational approach for attacking large, graph-based optimization problems. Successful applications of this mechanism include I/O subsystem performance enhancement through disk cylinder remapping [14, 13], file assignment in a distributed network to reduce disk access conflict [12], and new computer graphics techniques for digital halftoning [11] and color quantization [10]. The mechanism is based on maximum-entropy Gibbs measures, which is described in Reynold's dissertation [27], and provides a natural equivalence between Hopfield nets and the *simulated annealing* paradigm. This similarity allows you to select the method that best matches the problem at hand. For the SMT problem we will implement the

Simulated Annealing approach.

Simulated Annealing [22] is a probabilistic algorithm that has been applied to many optimization problems in which the set of feasible solutions is so large that an exhaustive search for an optimum solution is out of the question. Although Simulated Annealing does not necessarily provide an optimum solution, it usually provides a good solution in a user-selected amount of time. Hwang and Richards [19] have shown that the optimal placement of s Steiner Points to n original vertices yields a feasible solution space of the size

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

provided that none of the original points have degree 3 in the SMT. If the degree restriction is removed they showed that the number is even larger. The SMT problem is therefore a good candidate for this approach.

Adding 1 Junction

Georgakopoulos and Papadimitriou [15] have provided an $\mathcal{O}(n^2)$ solution to the *1-Steiner problem*, wherein exactly one Steiner Point is added to the original set of points. Since at most $n-2$ Steiner Points are needed in an SMT solution, repeated application of the algorithm offers a "greedy" $\mathcal{O}(n^3)$ approach. Using their method, the first Steiner Point is selected by partitioning the plane into Oriented Dirichlet Cells, which they describe in detail. Since these cells do not need to be discarded and recalculated for each addition, subsequent additions can be accomplished in linear time. Deletion of a candidate Steiner Point requires regeneration of the MST, which Shamos showed can be accomplished in $\mathcal{O}(n\ log\ n)$ time if the points are in the plane [26], followed by the cost for a first addition ($\mathcal{O}(n^2)$). This approach can be regarded as a natural starting point for Simulated Annealing by adding and deleting different Steiner Points.

The Heuristic

The Georgakopoulos and Papadimitriou 1-Steiner algorithm and the Shamos MST algorithm are both difficult to implement. As a result, we have chosen to investigate the potential effectiveness of this Annealing Algorithm using a more direct, but slightly more expensive $\mathcal{O}(n^3)$ approach. As previously noted, all Steiner Points have degree 3 with edges meeting in angles of 120°. We consider all $\binom{n}{3}$ triples where the largest angle is less than 120°, compute the Steiner Point for each (a simple geometric construction), select that Steiner Point giving greatest reduction, or least increase in the length of the modified tree (increases are allowed since the Annealing Algorithm may go uphill) and update the MST accordingly. Again, only the first addition requires this (now $\mathcal{O}(n^3)$) step. We use the straightforward, $\mathcal{O}(n^2)$ Prim's algorithm to generate the MST initially and after each deletion of a Steiner Point.

The Annealing Algorithm can be described as a non-deterministic walk on a surface. The points on the surface correspond to the lengths of all feasible solutions, where two solutions are adjacent if they can be reached through the addition or deletion of one Steiner Point. The probability of going uphill on this surface is higher when the temperature is higher but decreases as the temperature cools. The rate of this cooling typically will determine how good your solution will be. The major portion of this algorithm is presented in Figure 2. This non-deterministic walk, starting with the MST has led to some very exciting results.

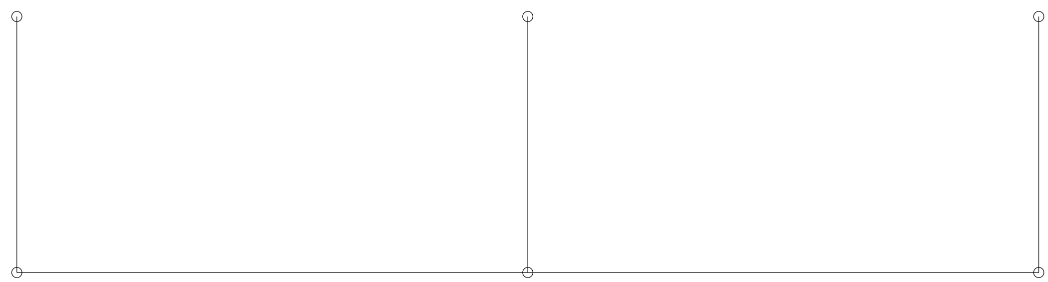
```
#define EQUILIBRIUM ((accepts>=100 AND rejects>=200) OR
     (accepts+rejects > 500))
#define FROZEN ((temperature < 0.5) OR ((temperature < 1.0)
     AND (accepts==0)))

while(not(FROZEN)){
     accepts = rejects = 0;
     old_energy = energy();
     while(not(EQUILIBRIUM)){
          operation = add_or_delete();
          switch(operation){
               case ADD:
                    ΔE = energy_change_from_adding_a_node();
                    break;
               case DELETE:
                    ΔE = energy_change_from_deleting_a_node();
                    break;
          }
          if(rand(0,1) < e^{min{0.0,-ΔE/temperature}}){
               accepts++;
               old_energy = new_energy;
          }else {
               /* put them back */
               undo_change(operation);
               rejects++;
          }
     }
     temperature = temperature*0.8;
}
```

Figure 2: Simulated Annealing Algorithm

5 Results

Before we discuss large problems, a simple introduction into the results from a simple six point problem is in order. The Annealing Algorithm is given the coordinates for six points: (0,0), (0,1), (2,0), (2,1), (4,0), (4,1). The first step is to calculate the

MST, which has a length of 7, as shown in Figure 3. The output of the Annealing Algorithm for this simple problem is shown in Figure 4. In this case the Annealing Algorithm calculates the exact SMT solution which has a length of 6.616994.



(0,0) (0,1) (2,0) (2,1) (4,0) (4,1)

Figure 3: Spanning Tree for 6 point problem.

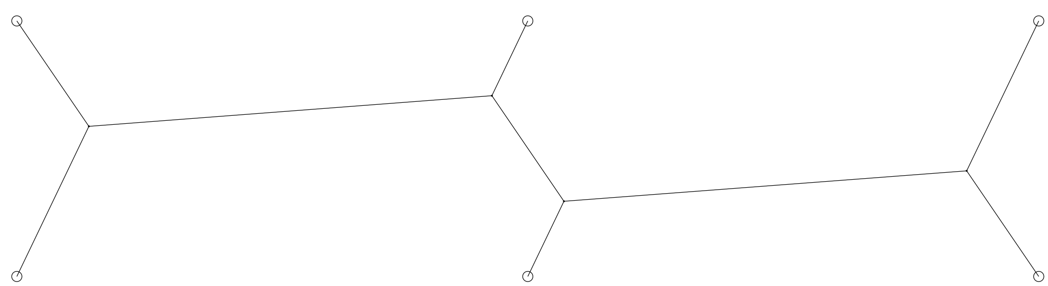


Figure 4: 6 point solution.

We propose as a measure of accuracy the percentage of the difference between the length of the MST and the exact SMT solution that the Annealing Algorithm achieves. This is a new measure which has not been discussed (or used) because exact solutions have not been calculated for anything but the most simple layouts of points. For the six point problem discussed above this percentage is 100.0% (the exact solution is obtained).

After communicating with Cockayne, data sets were obtained for exact solutions to randomly generated 100 point problems that were developed for [5]. This allows us to use the measure of accuracy previously described. Results for some of these data sets provided by Cockayne are shown in Table 1.

An interesting aspect of the Annealing Algorithm that cannot be shown in the table is the comparison of execution times with Cockayne's program. Whereas Cockayne mentioned that his results had an execution cut-off of 12 hours, these results were

Exact Solution	Spanning Tree	Simulated Annealing	Percent Covered
6.255463	6.448690	6.261797	96.39%
6.759661	6.935189	6.763495	98.29%
6.667217	6.923836	6.675194	96.89%
6.719102	6.921413	6.721283	99.01%
6.759659	6.935187	6.763493	98.29%
6.285690	6.484320	6.289342	98.48%

Table 1: Results from 100 point problems

obtained in less than 1 hour. The graphical output for the first line of the table, which reaches over 96% of the optimal value, appears as follows: the data points and the MST are shown in Figure 5, the Simulated Annealing Result is in Figure 6, and the Exact SMT Solution is in Figure 7. The solution presented here is obtained in less than $\frac{1}{10}th$ of the time with less than 4% of the possible range not covered. This indicates that we could hope to extend our Annealing Algorithm to much larger problems, perhaps as large as $1,000$ points. If we were to extend this approach to larger problems we would definitely need to implement the Georgakopoulos-Papadimitriou 1-Steiner Algorithm and the Shamos MST Algorithm.

References

[1] M.W. Bern and R.L. Graham. The shortest-network problem. *Sci. Am.*, 260(1):84–89, January 1989.

[2] E.J. Cockayne. On the Steiner problem. *Canad. Math. Bull.*, 10(3):431–450, 1967.

[3] E.J. Cockayne. On the efficiency of the algorithm for Steiner minimal trees. *SIAM J. Appl. Math.*, 18(1):150–159, January 1970.

[4] E.J. Cockayne and D.E. Hewgill. Exact computation of Steiner minimal trees in the plane. *Info. Proccess. Lett.*, 22(3):151–156, March 1986.

[5] E.J. Cockayne and D.E. Hewgill. Improved computation of plane Steiner minimal trees. *Algorithmica*, 7(2/3):219–229, 1992.

[6] E.J. Cockayne and D.G. Schiller. Computation of Steiner minimal trees. In D.J.A. Welsh and D.R. Woodall, editors, *Combinatorics*, pages 52–71, Maitland House, Warrior Square, Southend-on-Sea, Essex SS1 2J4, 1972. Mathematical Institute, Oxford, Inst. Math. Appl.

[7] R. Courant and H. Robbins. *What is Mathematics? an elementary approach to ideas and methods.* Oxford University Press, London, 1941.

[8] D.Z. Du and F.H. Hwang. A proof of the Gilbert-Pollak conjecture on the Steiner ratio. *Algorithmica*, 7(2/3):121–135, 1992.

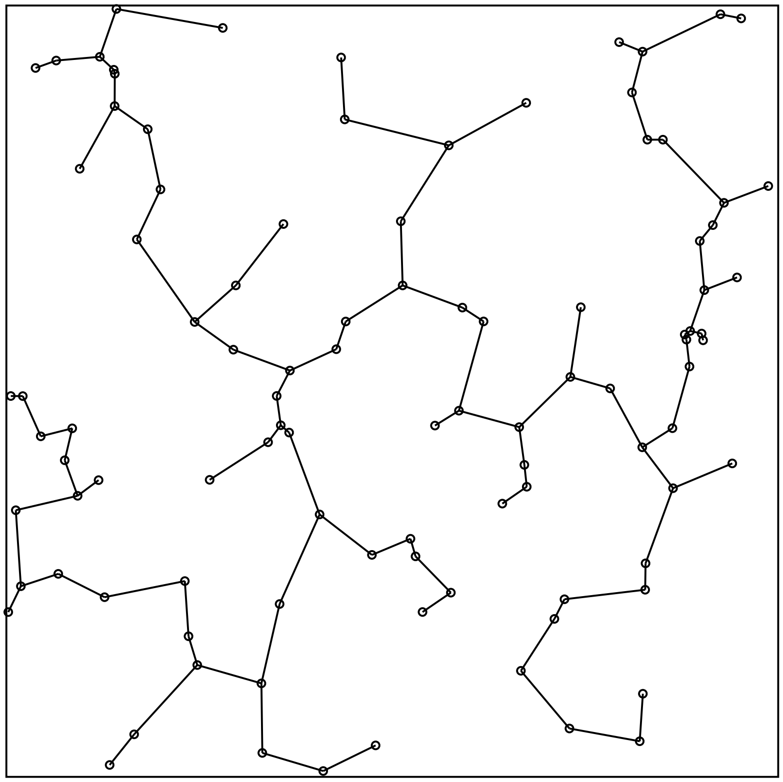


Figure 5: Spanning Tree

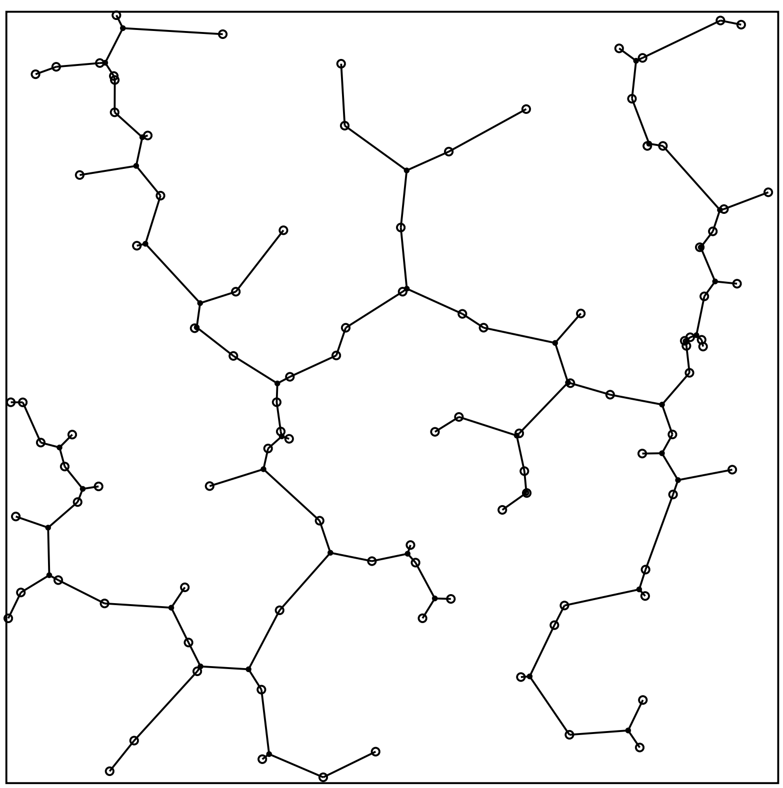


Figure 6: Simulated Annealing Solution

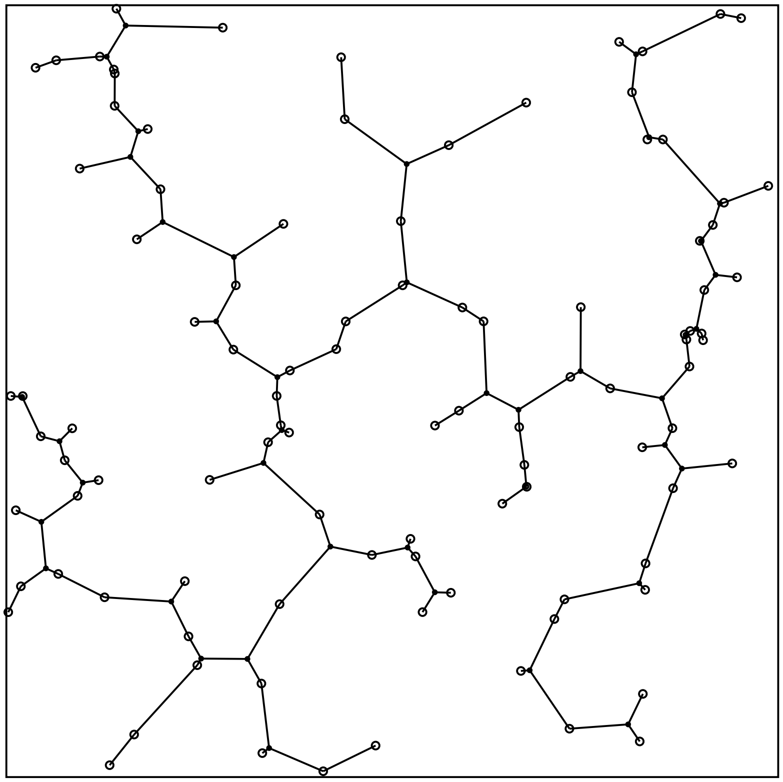


Figure 7: Exact Solution

[9] M.R. Garey, R.L. Graham, and D.S Johnson. The complexity of computing Steiner minimal trees. *SIAM J. Appl. Math.*, 32(4):835–859, June 1977.

[10] R. Geist, R. Reynolds, and C. Dove. Context sensitive color quantization. Technical Report 91–120, Dept. of Comp. Sci., Clemson Univ., Clemson, SC 29634, July 1991.

[11] R. Geist, R. Reynolds, and D. Suggs. A markovian framework for digital halftoning. *ACM Trans. Graphics*, 12(2):136–159, April 1993.

[12] R. Geist and D. Suggs. Neural networks for the design of distributed, fault-tolerant, computing environments. In *Proc. 11^{th} IEEE Symp. on Reliable Distributed Systems (SRDS)*, pages 189–195, Houston, Texas, October 1992.

[13] R. Geist, D. Suggs, and R. Reynolds. Minimizing mean seek distance in mirrored disk systems by cylinder remapping. In *Proc. 16^{th} IFIP Int. Symp. on Computer Performance Modeling, Measurement, and Evaluation (PERFORMANCE '93)*, pages 91–108, Rome, Italy, September 1993.

[14] R. Geist, D. Suggs, R. Reynolds, S. Divatia, F. Harris, E. Foster, and P. Kolte. Disk performance enhancement through Markov-based cylinder remapping. In Cherri M. Pancake and Douglas S. Reeves, editors, *Proc. of the ACM Southeastern Regional Conf.*, pages 23–28, Raleigh, North Carolina, April 1992. The Association for Computing Machinery, Inc.

[15] G. Georgakopoulos and C. Papadimitriou. A 1-steiner tree problem. *J. Algorithms*, 8(1):122–130, Mar 1987.

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

[17] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1:17–61, 1988.

[18] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci.*, 81:3088–3092, 1984.

[19] F.K. Hwang and D.S. Richards. Steiner tree problems. *Networks*, 22(1):55–89, January 1992.

[20] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem*, volume 53 of *Ann. Discrete Math.* North-Holland, Amsterdam, 1992.

[21] V. Jarník and O. Kössler. O minimálnich gratech obsahujicich n daných bodu [in Czech]. *Casopis Pesk. Mat. Fyr.*, 63:223–235, 1934.

[22] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(13):671–680, May 1983.

[23] Z.A. Melzak. On the problem of Steiner. *Canad. Math. Bull.*, 4(2):143–150, 1961.

[24] Michael K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Trans. Comput.*, C-31(9):913–917, September 1982.

[25] J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

[26] F.P. Preparata and M.I. Shamos. *Computational Geometry: an introduction.* Springer-Verlag, New York, NY, 1988.

[27] W.R. Reynolds. *A Markov Random Field Approach to Large Combinatorial Optimization Problems.* PhD thesis, Clemson, University, Clemson, SC 29634, August 1993.

[28] P. Winter. An algorithm for the Steiner problem in the Euclidian plane. *Networks*, 15(3):323–345, Fall 1985.