STEINER TREE PROBLEMS

F. K. Hwang

Dana S. Richards

Computer Science Report No. TR-87-21
November 17, 1987

STEINER TREE PROBLEMS

F. K. Hwang
Bell Laboratories

Dana S. Richards
University of Virginia

1. Introduction

The Steiner minimal tree problem was first studied for the euclidean distance metric. Given a set N of n points in the euclidean plane, the shortest network interconnecting N is called a *Steiner minimal tree* (denoted ESMT or simply SMT) for N. Junctions of edges in the network which are not points in N are called *Steiner points* or *S-points* ; the original points are *N-points*. The study of the euclidean SMT problem for the special case $n = 3$ has a long history back to 17th century with names like Cavalieri, Fermat, Simpson, Torricelli, Heinen, Viviani and Steiner among others (e.g., [119, 187]). However, the $n = 3$ Steiner problem is also a special case of what is called "generalized Fermat problem" by Kuhn [109], which asks for a point to minimize the sum of distances between that point and all points of N. Since only for $n \geq 4$ the SMT problem asserts its own character distinct from the generalized Fermat problem, we will consider the 1934 Jarnik and Kössler paper [98] as the first paper on the ''genuine'' SMT problem. However, the real awakening of interests in the SMT problem is due to its inclusion in the popular book *What Is Mathematics?* by Courant and Robbins [37] in 1941, who also attached the name ''Steiner'' to the problem.

There have been many generalizations of the SMT problem. Some have changed the metric and costs but have preserved the geometric basis. The rectilinear metric has received much attention due to its use in many applications, such as wiring in VLSI, and utilities in buildings. Some generalizations have abstracted the problem as finding the the minimum connecting network for a subset of all possible points; the graphical formulation is the primary example. Many other cases are treated below.

2. The Euclidean SMT Problem

Mechanical solutions of the SMT problem by using pegs and elastic bands, or soap films were nicely covered by Miehle [121]; see also Courant and Robbins [37, p. 392], Steinhaus [160, p. 119], Gallawa [68], Clark [28], Polya [129], and the excellent survey paper by Gilbert and Pollak [75]. Soukup [159] provides a solution using a rubber membrane. These techniques are representative of the class of ''analog'' techniques that use natural systems' tendencies to find minimum energy configurations to solve optimization problems. However, our survey will concentrate on mathematical analyses and solutions to the SMT problem.

An interconnecting network T is called a *Steiner tree*, ST, if it satisfies the following three conditions

(i) T is a tree.

(ii) Any two edges of T meet at an angle of at least $120°$.

(iii) An S-point cannot be of degree one or two.

Clearly, a shortest interconnecting network must be an ST. Note that (ii) implies that any point of T can have degree at most three. (ii) and (iii) together imply that each S-point is of degree exactly three. (i) and (iii) together imply that there are at most $n-2$ S-points. An ST is *full* (denoted FST) if it contains $n-2$ S-points. It is easily verified that each N-point is of degree one in an FST. The graph structure, P, of an ST is called a *topology*. A *full topology* (FP) is the topology for some FST. By a convexity argument it is easily proved that a topology P can have at most one ST; when the ST exists, it is the shortest interconnecting network for P.

2.1. Exact Algorithms

Melzak [118] first gave a finite algorithm for constructing SMT for general set N; also see Cockayne [30]. The general approach is to partition each topology P into full topologies $\{FP_i\}$ and apply what Melzak called a "euclidean construction" to obtain an FST on each FP_i. If each FP_i has an FST, then the union of these FSTs yields the ST for P. If any of the FP_i does not have an FST, then neither does P have an ST. Compare the lengths of the STs of all possible topologies and select a shortest one to be the SMT for N.

Melzak's algorithm takes exponential time. There are two sources of exponentiality, one minor and one major. The minor one is due to the euclidean construction which, at each step, chooses one of two possible substitution points (as there are two equilateral triangles for a given side). Since the correctness of the choice is not known only until an ST is either constructed or demonstrated to be nonexistent, backtracking is necessary which requires a worst-time complexity of $O(2^s)$ where s is the number of S-points in the given full topology. Hwang [91] recently removed this minor source by showing that when proceeding with a specified sequence of S-points, each substitution point can be chosen correctly at each step. Thus no backtracking is necessary and the time complexity is linear.

The major source of exponentially is due to the large number of topologies. If no N-point is allowed to have three edges, the number is $2^{-n}\begin{bmatrix} n \\ s+2 \end{bmatrix}\frac{(n-s-2)!}{s!}$, for n N-points and s S-points. The number is even bigger when this restriction is dropped. For example, this number is 5625 for $n=6$, 110880 for $n=7$, and 2643795 for $n=8$. Many pruning techniques for quickly eliminating the FPs have been proposed (see [75] for a good collection), but they do not affect the exponentiality. The two most effective techniques are:

(i) *A cyclic order of N-points.* Given an FST, Cockayne [32] defined a cyclic order of the N-points by always making a left turn, after leaving an N-point, to reach the next N-point (note that Werner [183] defined a different cyclic order by alternating left and right turns). While it is well known [75] that an SMT must lie in the convex hull of N, Cockayne extended its validity to the "Steiner hull", obtained from convex hull by sequentially removing triangles. Cockayne showed that the clockwise order of N-points on the Steiner

polygon, which bound the Steiner hull, is consistent with the cyclic order he defined. We will call a topology *oriented* if the cyclical order of the three edges incident to each S-point is also given. Then the ordering of N-points on the Steiner polygon not only eliminates many topologies, but also topologies with wrong orientations.

(ii) *Restriction on S-points.* It is well known [75] that an S-point s adjacent to two points a and b must lie on the 120° arc of the circle $C(a, b)$ which circumscribes an equilateral triangle with $[a, b]$ as one side. Now if a (b) itself is an S-point confined to an arc, the arc confining s can be made smaller than 120° since it can cross neither the arc of a nor the arc of b. Winter [188] showed that for most full topologies a few rounds of nested confinement will reduce the arc of an S-point to an empty arc, thus eliminating the topology.

Melzak's algorithm has provided the core of many computer programs. The first published version was by Cockayne and Schiller [33], called STEINER, which can solve cases as large as a 7-point problem. Boyce and Seery [13] and Boyce [14] improved it to allow solutions of up to 10 points, mostly by using a complex multiplication to compute the substitution points. Cockayne and Peters' [34] MFST82 contained a few more geometric tests to eliminate certain cyclic orders. Winter [188], by using restriction (ii) as well as some criteria for the nonoptimality of an FST, was able to solve problems of up to 15 points in his GEOSTEINER program. He noted that for $n \geq 15$ the computation time needed to form the unions of "surviving" FSTs dominates the computation time needed for the construction of FSTs. Cockayne and Hewgill [35] improved the first aspect and raised the "solvable" range to 30 points. An algebraic approach for the euclidean construction, which should ease the calculations in these programs, was recently proposed by Hwang and Weng [92]. They need hexagonal coordinates so that the coordinates of an S-point is readily unavailable once the coordinates of its three adjacent points are known, provided the hexagonal coordinate system is given by the three directions of the SMT. They showed that starting from a given hexagonal coordinate system, the transformation to the SMT-defined system can be easily found.

The exponentiality is more a problem inherent with the SMT, rather than just with Melzak's algorithm. Garey, Graham and Johnson [70] have shown that the SMT decision problem is NP-complete. For NP-complete problems "decomposability" is often an important consideration. For the SMT problem *decomposability* means the partitioning of N into subsets $N_1, \cdots, N_m$ such that $N = \bigcup_{i=1}^{m} N_i$, $|N_i \cap N_j| \leq 1$ for all i, j, and the union of the SMTs for each N_i is an SMT on N. There are three decomposition theorems so far. The first is due to the "double wedge" property as given by Gilbert and Pollak [75]. The second is a consequence of the "Steiner polygon" intersecting itself as given by Cockayne [32]. The third is a recent result of Hwang, Song, Ting and Du [93] who gave conditions under which a quadrilateral $ABCD$, where (A, B) and (C, D) are two pairs of adjacent points on the Steiner polygon, can be removed to decompose the Steiner polygon into two smaller polygons linked by an edge.

We close this section by mentioning some works on SMT which do not fit in the above discussions. Few [58] used a clever argument to show that the length of an SMT for n random points in a unit square is upper bounded by $\sqrt{n} + 7/4$. Chung and Graham [25] improved it to $.995\sqrt{n}$. Megiddo [117] showed that a game between the N-points, where the cost of an SMT is to be fairly split, can have an empty core, contrary to the minimal spanning tree

case.

2.2. Heuristic Algorithms

An important consideration for NP-complete problems is the existence of efficient heuristics with provably good performance. A minimal spanning tree (MST) algorithm can be thought of as a heuristic for SMT, and can be implemented in $O(n \log n)$ time as first demonstrated by Shamos and Hoey [148]. Define the *Steiner ratio* as

$$\rho = \min_{N} \frac{\text{length of an SMT on } N}{\text{length of an MST on } N}$$

Then ρ is a measure of how good the minimal spanning tree is. Gilbert and Pollak [75] conjectured $\rho = \sqrt{3}/2$ and proved it for $n = 3$. Later, Pollak [128] proved it for $n = 4$ (see also Du, Yao, and Hwang [44]), and Du, Hwang and Yao [47] proved it for $n = 5$. For general n, the lower bound for ρ has been steadily pushed up from .5 (Moore as reported in [75]) to .57 (Graham and Hwang [76]) to .74 (Chung and Hwang [23]), to .8 (Du and Hwang [45]) through geometrical arguments. Recently, Chung and Graham [26] obtained $\rho \geq .824$ by computer-aided calculations.

Chang [18] and Thompson [168] (also see Korhonen [105]) first had the idea of converting a minimal spanning tree to an ST through a sequence of "local steinerizations", i.e., add an S-point to connect three points "close" to each other and delete an edge from any loops formed. This idea has been the backbone of many heuristics for both the euclidean and the rectilinear distance metrics, though the exact definition of local steinerization differ. Chang and Thompson did not tailor their algorithms from the computational complexity viewpoint since it was not fashionable then. A literal implementation of Chang's algorithm requires $O(n^4)$ time. Smith and Liebman [151] gave a similar but more complex $O(n^4)$ time algorithm. Smith, Lee and Liebman [152] gave a $O(n \log n)$ algorithm that used Voronoi diagrams and Delauney triangulations. The improvement using these heuristics over a minimal spanning tree is typically 3-4% in expected length; but no improvement in the worst-case has been demonstrated.

A completely different heuristic using "simulated annealing" was proposed by Lundy [114, 115]. Such an approach will successively perturb candidate SMTs searching for an (locally) optimal solution. The perturbation scheme employed replaces three edges [s,1], [s,2], [3,4] by the three edges [s,3], [s,4], [1,2] where s is an S-point. Note that such a scheme not only improves the embedding of a topology but explores many different topologies. Some experimental results were given.

Finally, Soukup proposed a scheme motivated by a physical analog approach [159]. It involves a root-finding step for a derived polynomial.

2.3. Special Cases

Another important avenue of research with NP-complete problems is the study of special-cases. By using the special geometry of the given set N, one may obtain an explicit construction or a fast algorithm for an SMT. The first special set N is the set of vertices of a regular n-gon as studied in the first genuine SMT paper of Jarnik and

Kössler [98]. They obtained the SMT for $3 \le n \le 5$ and also proved that for $n = 6$ or $n \ge 13$ the SMT is the perimeter of the n-gon minus a side. Recently, Du, Hwang, and Weng [50] (see also Kotzig [106] and Weng [181]) finished the problem by proving the same for $7 \le n \le 12$. When N is a set of vertices on a circle with radius r, Du, Hwang and Chao [48] proved that the SMT is the perimeter of the n-gon minus a longest side when at most one side can be longer than $.557r$. Curiously, the result depends crucially on the value of a lower bound of ρ.

Other special sets N whose SMTs can be efficiently constructed are ladders (Chung and Graham [22]) and their generalizations (Du and Hwang [52]), vertices on a zigzag line (Du, Hwang and Weng [46]), and vertices on a splitting tree (Hwang, Weng and Du [90]). We can also restrict N by fixing n. As mentioned in the opening paragraph, the $n = 3$ SMT was well studied by 17th century mathematicians. The $n = 4$ case was studied by Pollak [128], Ollerenshaw [125], Du, Yao, and Hwang [44], and Du, Hwang, Song and Ting [51]. It should be noted that a good understanding of the $n = 3, 4$ cases are crucial in obtaining two of the decomposition results, one by removing a triangle and the other a quadrilateral.

2.4. Generalizations

So far we have only discussed the euclidean SMT problem on the plane. For higher dimensional euclidean space Gilbert and Pollak [75] showed that most of the basic properties of SMT still hold. The extension of the euclidean construction of an FST to d dimensions is nontrivial and has not been attempted. The Steiner ratio was conjectured [75] to be achieved by the corners of the d-dimensional simplex. However, SMTs for d-dimensional simplices are not known in general, Chung and Gilbert [21] gave an upper bound for their lengths which approaches .669842 for large d.

Four other types of generalizations have been attempted for the euclidean SMT problem. The first is a generalization of N from a point-set to other objects, as well as adding other objects in the space as obstacles. The second is a generalization of the euclidean space to some other metric space. The third is Gilbert's flow-dependent networks. And the fourth is a generalization on the connection property.

For the first type, Cockayne and Melzak [29] extended Melzak's algorithm to the case that N is a set of point-sets. Weng [182] used the hexagonal coordinate system to treat the case that N is a set of regions. Trietsch [171] gave a finite-time algorithm for adding points to an existing network. The SMT problem with obstacles is a practical problem in pipeline layout or routing heating mains; see Smith and Liebman [151], Liestman [113], and Smith [156]. Provan [131] has given effective approximation algorithms for obstacles for the case where the tree is confined to a (not necessarily simply) connected polygonal region.

For the second type, Soukup [20, 158] studied nonlinear cost and proved that if the cost function is increasing in a certain sense, then the degree of each vertex is bounded. He also proved that if the cost function is convex, then an SMT is full. Sankoff and Rousseau [141] gave a dynamic programming solution for locating S-points of an FST in an arbitrary metric space in their study of phylogeny.

For the third type, Gilbert [74] first studied the shortest interconnecting network which must satisfies a given matrix of two-way flows, $f(i, j)$, where the cost of an edge with capacity c is $g(c)$ per mile. When $g(c)$ is a constant the shortest network is an SMT. But in general, the shortest network loses many of the properties of an SMT and need not even be a tree. This network was also studied by van de Heyden [174], Trietsch and Handler [170], and Du and Hwang [49]. Werner [183] considered a flow matrix which affects only the usage, but not the size, of an edge. Under a linear cost model he showed that if the flow demand decreases at least linearly with the distance the minimum cost network is an SMT.

The fourth type, directed SMTs, were studied by Lee [110] for drainage networks and by Bhashkaran [11] and Bhashkaran and Salzborn [12] for gas pipeline networks. In these networks flows are directed from a set of sources to a designated sink. The latter work proved that the optimal gas-pipeline network must be a tree under a fairly general set of conditions. Smith [150] formulated it as a "directed steiner minimal tree" problem. Nodes have weights which can be interpreted as source flows. The resulting Steiner tree is "directed" by the flow interpretation and the edge weights are a function of the flow and its direction. Several different weight function can be used. Sandifer [140] gave a heuristic for one weight function (CUMSUM).

As mentioned in the introduction the $n = 3$ case has been alternately generalized as the "Fermat problem", where there is only one additional point and all N-points connect to that point. If each N-point has a weight and the lengths of the edges to the central point are biased by the weights, then the minimization problem is the "Weber problem". This is the cornerstone of a vast literature for Location Theory. Some interesting work has been done on this problem, often misattributing the problem as a Steiner problem, e.g., [31, 100, 103, 120, 126, 169]. An $O(n^2)$ time algorithm is known [72] for the unit weights case.

3. The Graphical SMT Problem

The graphical Steiner minimal tree (GSMT) problem asks for a minimum tree spanning some designated vertices of a graph. Formally, a graph $G = (V, E)$ is given, with weights on the edges, and a set $N \subseteq V$ is specified. A solution to the GSMT problem is the minimum weight tree subgraph of G that spans N. The vertices not in N with degree ≥ 3 in the tree are called *Steiner vertices*. (A more general formulation asks for a minimum weight subgraph, perhaps cyclic. If the edge weights are nonnegative the two formulations are essentially the same. Otherwise a simple preprocessing step can be used [79] to handle the negative edges.) Hakimi [79] and Levin [112] independently posed this problem. While it is possible to motivate the graphical case from the euclidean case, the two problems have little in common. Recall, however, that some heuristics for the euclidean case have graphical origins.

Research on the GSMT problem has taken a familiar route. First, some exponential time exact algorithms for finding the optimal configuration were posed. Then the GSMT problem and its variants were shown to be NP-complete. Finally, many heuristics and special cases have been considered. Winter has presented an excellent survey [191].

3.1. Exact Algorithms

Hakimi [79] proposed an exact algorithm that ran in $O(2^{v-n} v^2 + v^3)$ time, where $v = |V|$, and $n = |N|$. This algorithm tries all possible choices for the Steiner vertices and for each choice computes the corresponding minimum tree. Levin [112] proposed a dynamic programming approach (i.e., precomputing possible subtrees) that ran in $O(3^v n + 2^v n^2)$ time. Essentially the same algorithm was given by Dreyfus and Wagner [43] with a time complexity of $O(3^v n + 2^v n^2 + n^3)$. A branch-and-bound approach has been proposed [60, 149] that uses heuristics to provide good lower bounds and to choose the next edge for consideration in the backtracking process. Of course, it is difficult to analyze this algorithm but empirical evidence [61] suggests that it compares favorably to the other two when n is about $\frac{1}{2} v$; Hakimi's algorithm is favored when n is about v and the dynamic programming approach is favored when n is small.

Other exponential time exact algorithms have also appeared without analyses. Hakimi [79] proposed a recursive approach to enumerate the various tree topologies, and gave a useful theorem to prune the search. Balakrishnan and Patel [4, 5] gave an algorithm that cleverly generates spanning trees for a derived problem in order of increasing weight until a solution to the original problem can be inferred. Aneja [3] reformulated the problem as a set covering problem, presented as a 0-1 linear programming problem. Several relaxation techniques and improvements were used. Beasley [6] gave a branch-and-bound algorithm where the bounds used Lagrangean relaxations of several 0-1 linear programming reformulations; see also [53]. Further constraints were added [7] leading to improved performance. In fact, the latter algorithm has the best reported performance on large random sparse graphs; with $v = 500$ and $n = 125$ less than 6 minutes were required on some sparse graphs. Another 0-1 linear programming algorithm was given by Wong [195] that combined two known approaches from cognate problems. This formulation admits directions on the edges and hence applies to directed graphs, discussed below. Segev [146] used similar techniques for a weighted variant of the GSMT problem. Jain [97] used probabilistic arguments to show that a linear programming relaxation (of an aggregate formulation) can give poor bounds for the integer solution.

Algorithms for the GSMT problem can benefit from processing the graph, especially those that incrementally build the SMT. There are several obvious reductions that can be made (e.g., handling vertices of degree 1) as well as some nontrivial reductions (e.g., using triangle inequalities when possible.) These are discussed by Beasley [6], Balakrishnan and Patel [5], and Iwainsky et al. [95]. In general there are no known theorems bounding the length of a GSMT, beyond the results discussed in the heuristics section below.

An early result, due to Karp, was the GSMT decision problem was NP-complete [101, 102]. Many restricted versions remain hard. If G is planar the problem is still NP-complete [69]. Further, Berlekamp showed that the GSMT remains NP-complete if all edges have unit weights, even if G is bipartite with all edges from N to $V - N$ [71, p.209]. The problem remains NP-complete for chordal graphs and split graphs [184]. (Hakimi [79] gave several explicit problem reductions that are implicit for every NP-complete problem.) Hence there has been much interest in heuristics and special cases.

3.2. Heuristic Algorithms

Many heuristics have been proposed for finding approximate solutions to the GSMT problem. Those in the first group are based on known graph algorithms, usually for the minimum spanning tree (MST) problem. A second group of algorithms use suboptimal variants of exact algorithms. The latter group are not necessarily polynomial-time.

A straightforward MST-based algorithms has been proposed by El-Arbi [54], and independently by Plesnik [127], Kou, Markowsky, and Berman [107], and Iwainsky et al. [95]. The algorithm begins by constructing a complete graph G' on the vertex set N with edge weights equal to the lengths of the corresponding shortest paths in the original graph G. Then an MST is found for G' and each edge of that MST corresponds to a shortest path in G. Let G'' be the subgraph of G that is the union of these paths. The approximate GSMT is the MST of G'', after some obvious postprocessing. Let L_{MST} and L_{OPT} be the length of the approximate GSMT found and the optimal GSMT, respectively. It is known [107] that $\frac{L_{\text{MST}}}{L_{\text{OPT}}} \leq 2(1-\frac{1}{t})$, where the optimal tree has t leaves; see also [54, 95, 127]. These algorithms run in $O(n v^2)$ time, where the construction of G' dominates. Variants are discussed in [127].

Wu, Widmayer, and Wong [196] propose an implementation, based on Kruskal's MST algorithm, that does not explicitly construct G'. Recall, Kruskal's algorithm begins with v singleton trees and iteratively connects the two nearest current trees, until one tree remains. To avoid constructing G', the algorithm simultaneously grows "shortest path trees" from each vertex (always adding the least unused edge to one of these trees). When two trees meet a path between their roots is formed, corresponding to the next edge Kruskal's algorithm on G' would have used. The algorithm runs in $O(e \log v)$ time, where $e = |E|$, and has the same performance bounds as above. A similar algorithm was independently given by Wang [180]. Widmayer [185] compared all the above algorithms and showed that no one is always superior to the others.

Widmayer [186] refined the above algorithm (to better handle difficult cases) and used an efficient priority queue data structure, the Fibonacci heap. The resulting algorithm has the same performance bounds and runs in $O(e + (v + \min\{e, n^2\}) \log v)$. This algorithm is never (asymptotically) slower than the above algorithms and can be faster. It has been shown [108] that the Kruskal-based approach is (essentially) optimal for random unweighted graphs, i.e., where each edge is present with probability p. In that case the expected size of the GSMT is approximately $n \frac{\log v}{\log vp}$. (The trivial algorithm of using a shortest-path spanning tree is also nearly optimal for random graphs!)

Takahashi and Matsuyama [163] gave a heuristic based on Prim's MST algorithm. Recall, Prim's algorithm begins with one singleton tree and iteratively connects the nearest vertex to it, until there is one tree. Their algorithm begins with one vertex of N and connects, with a path, the nearest other point in N. At each step there is some tree and the nearest unused vertex in N is connected, with a path, to some vertex in the tree. The performance bound is $\frac{L_{\text{PRIM}}}{L_{\text{OPT}}} < 2(1-\frac{1}{n})$, where L_{PRIM} is the length of the approximate GSMT. The algorithm runs in $O(n v^2)$

time. A variant has been proposed that has the same performance bounds and time complexity but uses less space [94]. The algorithm was independently given by Wang [180] and it is related to the heuristics above [135]. Segev [146] and Ma [116] have given simple Prim-based algorithms.

Rayward-Smith's heuristic [64, 134, 135] explicitly selects the Steiner vertices, while the above approaches implicitly choose them. It is Kruskal-based but does not connect the two nearest subtrees. Instead two trees may be connected to a Steiner vertex, chosen because it minimizes an "average distance" to the set of all current subtrees. It runs in $O(v^3)$ time and has good empirical performance [135].

Chen's two heuristics [19] are based on a subroutine for the case of $n = 3$. (We note that other special cases of $n = 2$ and $n = v$ are trivially handled by shortest-path and MST routines, respectively.) An exact $O(e \log v)$ time algorithm is given for this case; if the edges have unit weights then $O(e)$ time suffices. The first heuristic uses the routine to introduce "local steinerizations" into a Prim-based approach. The second heuristic generalizes Kruskal's algorithm by linking together three subtrees at each step. The time complexities are $O(\Delta n e \log v)$ and $O(n^2 e \log v)$, respectively, where Δ is the maximum degree.

Plesnik [127] presented another heuristic based on the identification of neighborhoods which can be contracted to a point. Before contraction the GSMT problem is solved within each neighborhood, and then the problem is recursively solved on the contracted graph. Two trivial heuristics have been proposed [163] and shown to have poor worst-case performance: simply pruning an MST of G and pruning a shortest-path spanning tree of G.

Several suboptimal variations of exact algorithms have been proposed. Aneja [3] gave a greedy variation of his set-covering approach. Wong [195] used an MST routine as a short-cut in his exact algorithm. Segev [146], in his work on a generalized GSMT problem, used an MST routine, as well as proposing a subgradient heuristic. Computational experience with all of these has been encouraging.

For related results see [96, 161, 165, 166, 173].

3.3. Special Cases

Polynomial time algorithms can be found for classes of graphs where enough structure ensures restrictions on possible GSMTs. It is discouraging that the problem remains hard for planar graphs; in fact for grid graphs (see the next section). However the problem can be solved efficiently for outerplanar graphs and other classes where there are strong decomposition theorems. Related results are discussed in the next section.

Perhaps the first result was by Wald and Colbourn [175] for outerplanar graphs. They generalized the result to the related classes of 2-trees, partial 2-trees, minimum IFI networks and their applications [176-179]; see also [130]. Series-parallel graphs have similar decomposability characterizations. Takamizawa et al. [164] indicated an algorithm for series-parallel graphs; see also [36, 133]. Winter has similar results for Halin networks [194]. All these algorithms run in linear time. White et al. [184] gave an $O(v^3)$ time algorithm for strongly chordal graphs with unit edge weights (but the problem remains NP-complete for strictly chordal graphs with strictly triangular weight functions).

Planar graphs can be solved in polynomial time if the vertices of N are found on just k faces of some given embedding of the graph, i.e., it is *k-planar*. (Bienstock and Monma have polynomial-time algorithms for finding optimal embeddings.) The k-planar case was independently solved by Erickson, Monma, and Veinott [55] and Bern [10] using the dynamic programming techniques of Dreyfus and Wagner. The time complexities are $O(v^{3k})$ and $O(v^{2k+2} + n^2 v^{2k})$, respectively. Provan [132] generalized these results by casting k-planarity as a specific "convexity" property.

3.4. Generalizations

There have been three principle avenues of generalization: changing the definition of the Steiner network, adding vertex weights, and using directed graphs.

Krarup (in an unpublished 1978 memo) suggested finding the minimum-weight 2-connected (or 2-edge-connected) subgraph that spans N in G, instead of a tree. Winter has given polynomial-time exact algorithms for outerplanar graphs [189], series-parallel graphs [190], and Halin networks [192]. In the latter paper the 3-connected case was solved. Duin and Volgenant [53] consider Steiner forests of at most m components. Cornuejols et al. [36] consider the "Steiner traveling salesman problem". Chung et al. discussed "Steiner caterpillars" [27] in connection with the analysis of a search problem.

Segev [146] proposed using vertex weights in addition to edge weights. He concentrated on the Single Point Weighted Steiner Tree problem, of which the GSMT problem is a special case. All vertex weights are negative (or nonpositive) and there is a distinguished vertex s. The desired tree includes s and is of minimum total weight. The formulation is motivated by having edge weights be the cost of establishing connections (to s) and the vertex weights are the payoffs. Exact and heuristic approaches are given. The formulation is actually in terms of directed graphs. Duin and Volgenant [53] discussed the general problem, where an s is not specified

Smith [150] has worked on a "directed steiner minimal tree" problem as described in section 2.4. In related work, Smith and Liebman [153] gave a heuristic based on local steinerizations of a rooted spanning arborescence, where the local updates use a Weber point instead of a Steiner point.

Wong [195] used a directed graph model with a distinguished vertex s, and asked for a minimum-weight directed tree, rooted at s, that spans N. It was cast as a multi-commodity network flow problem. He gave exact and heuristic algorithms, both discussed above; see also [53]. Suurballe [162] briefly mentions similar techniques. Nastansky et al. [123] use a similar model but restrict their attention to rooted directed acyclic graphs, with root s. They give an exact enumerative scheme that can be time-limited to give suboptimal answers. Their model is largely motivated by a m-dimensional grid graph with all edges directed away from the origin s; Trubin [172] also discussed this case.

Wald and Sorenson [178] consider symmetric directed graphs and do not specify the root of the Steiner tree. Further they may require some edges be used. They only present an exact algorithm for 2-trees. The problem is motivated by query inference applications for databases.

4. The Rectilinear SMT Problem

Rectilinear Steiner minimal trees (RSMT) are related to both the euclidean and the graphical cases. As with the euclidean case the basic problem has the n points of N located in the plane, and additional Steiner points may be used in constructing the minimal length tree. The distance metric is the rectilinear or Manhattan metric, i.e., the distance from a to b is $|x_a - x_b| + |y_a - y_b|$.

The first investigation of the RSMT problem was by Hanan [81]. He established the following important "dimension reduction" result. Extend horizontal and vertical lines through each of the points. Define the graph $G_N(V, E)$ by letting V be the set of intersections of these lines and there is an edge between two vertices if they are directly connected by a line, horizontally or vertically. Hanan showed that an RSMT is contained in G_N, i.e. the segments of the tree are composed of edges of G_N. An approach for generating candidate RSMTs has been given [124].

4.1. Exact Algorithms

While any exact algorithm for the GSMT problem could be used, only one exact algorithm for the RSMT has been proposed. Yang and Wing gave a straightforward branch-and-bound algorithm [198] and improved it somewhat [200]. It appears to be applicable only for $n < 10$. Sankoff and Rousseau [141] give a dynamic programming approach for the case when the topology is fixed. Thomborson et al. [167] give a $n^{O(\sqrt{n})}$ time algorithm that computes the optimal RSMT with high probability, when the points are uniformly distributed in the unit square.

The RSMT decision problem is NP-complete [69]. Therefore there has been much interest in heuristics and special cases.

4.2. Heuristic Algorithms

As with the euclidean SMT problem, a minimum spanning tree algorithm is a simple heuristic for the RSMT. The rectilinear minimum spanning tree (RMST) is defined in the obvious way, and can be found with any MST algorithm. (An embedding of the edges of an RMST may "overlap".)

Hwang [86] established many characterizations of RSMTs that allow their form to be sharply limited without loss of generality. For example, we may assume all tree edges consist of at most one horizontal and one vertical segment. He was able to establish

$$\frac{L_{\mathrm{RMST}}(N)}{L_{\mathrm{OPT}}(N)} \leq \frac{3}{2}$$

where $L_{\mathrm{RMST}}(N)$ is defined to be the length of the RMST and $L_{\mathrm{OPT}}(N)$ is the length of the RSMT. Empirically, evidence for random data suggests the expected value of the ratio is 1.13 [139, 199].

The expected length of an RSMT can be bounded. An early result is for RMSTs from which a similar bound follows for RSMTs. Let RMST(n) be the random variable for the length of the RMST for n points drawn uniformly

from the unit square. Gilbert [73] established that $E[\text{RMST}(n)] = O(\sqrt{n})$. Chung and Graham [25] give a sharper result;

$$\sqrt{n} + O(1) \leq F(n) \leq \sqrt{n} + 1 + o(1)$$

where $F(n)$ is the length of the longest possible RSMT on any n points drawn from the unit square. Komlos and Shing [104] showed that $\text{RSMT}(n) \geq \sqrt{n} / 5$ with probability $1 - o(1)$, where RSMT(n) is the random variable for the length of the RSMT for n points drawn uniformly from the unit square. While an RMST can serve as a good approximation to the RST, Chung and Hwang [24] showed that the semiperimeter of the least bounding rectangle, an often-used approximation, can be worse by a factor about $\frac{1}{2}\sqrt{n}$. (Such an approximation is optimal if $n = 3$ [81].)

The first nontrivial heuristic was proposed by Hanan in an unpublished report [80]. It found a tree using a Prim-based line-sweep algorithm, i.e., it built the tree in a greedy fashion from left to right. Four orthogonal sweeps were suggested. The best implementation seemed to require $O(n^2)$ time (e.g., [87]). Servit [147] suggested a simplified approach, which did not optimize each greedy step, that admitted an $O(n \log n)$ time implementation. Richards [136], using data structures used in Computational Geometry, was able to implement Hanan's full algorithm in $O(n \log n)$ time. The trees produced are slightly inferior to those of later heuristics. These approximate RSMTs are 4% better than the RMST, while later heuristics are usually 7% to 9% better. The six point-sets given by Soukup and Chow [157] have been extensively used for comparing heuristics.

Fu [67] gave an unanalyzed manual technique to iteratively improve a spanning tree by creating and breaking cycles. Hanan [82] disproves Fu's claim of optimality. An $O(n^4)$ time solution that uses several ad hoc stages was proposed [151]. It began by selecting a linear-sized subset of the n^2 vertices of G_N as candidates for Steiner points. Then it evaluated each in isolation and entered them into a priority queue. The queue determined the order the candidates were tried.

Yang and Wing proposed a suboptimal branch-and-bound algorithm which, while still exponential time, appears to be applicable for $n < 30$. When compared with known exact results the answers were remarkably close to optimal [199-201]. It simply uses Prim's algorithm but instead of choosing one of the two possible one-bend orientations of the new wire it explores both.

Several heuristics begin with an RMST. Since the underlying complete graph has $\Omega(n^2)$ edges, to improve on the $O(n^2)$ time bound requires preprocessing to reduce the size of the underlying graph. The rectilinear Voronoi diagram (defined analogously to the euclidean case) has a Delauney triangulation which contains enough edges to find the RMST. Since the triangulation corresponds to a planar graph the RMST can be found in $O(n)$ additional time. Hwang [89] shows, with standard divide-and-conquer techniques, how to find the Delauney triangulation in $O(n \log n)$ time . It should be noted that the details for the rectilinear Voronoi diagram are more complicated than for the euclidean case.

Hwang gave a heuristic for the RST problem, using the above RMST solution, that was based on earlier work by Lee, Bose, and Hwang [111]. That work built an RSMT in Prim fashion. It used a "3-point connection scheme" instead of simply connecting the nearest unused vertex. This involved a constant time search around the intended 2-point connection for three points which could be connected in a Steiner fashion, perhaps introducing a new Steiner point. It ran in $O(n^2)$ time with most of the time spent on deciding which point to connect next. Hwang [88] proposed an $O(n \log n)$ implementation that began with an RMST and from that inferred an ordering of the vertices which the above algorithm could use to decide which node to connect next and where to try to connect it.

Smith, Lee, and Liebman [154] proposed an $O(n \log n)$ time approach based on iteratively improving the RMST found over the Delauney triangulation. The technique is complex and used 4-point steinerizations as well. A simplified approach was proposed by Richards [136].

Bern and de Carvalho [8] investigated Kruskal-based approaches attributed to Thompson. Variations attributed to Ng and the themselves were proposed that were supposed to be faster than, but usually inferior to, the original. Thompson began with n singleton trees and at each stage a new wire connects two trees and the shortest such wire is chosen. This wire may not necessarily connect terminals but it can be "slid" so that it at least has one endpoint at a terminal, Steiner point, or corner of a previous wire. There are only $O(n)$ such positions. They assume the points are grid points of an $m \times m$ grid. They give an unusual analysis; even though m is theoretically unrelated to n (except $m > \sqrt{n}$) they assume a data structure with $O(m^2)$ is permissible. Thompson's algorithm is implemented in $O(m n^2 \log n)$ time and their variation takes $O(m n^2)$ time. Their algorithms can, by a small alteration, be made to run in $O(n^2 \log n)$ time [136].

Bern [9] has shown that the expected improvement of their approximate RSMT relative to the optimal RMST is bounded away from 0 in the limit. The proven bound, 0.00098, is much less than empirical studies indicate it should be. It is also shown that a linear number of Steiner points are expected. The results are for n points drawn at random from the unit square distributed according to a Poisson process with intensity n; a uniform distribution remains to be analyzed.

Komlos and Shing [104] proposed a divide-and-conquer algorithm based on two-dimensional partitioning. They start with n points, assumed to be uniformly distributed in the unit square, and a parameter t. Next iteratively partition, using medians, the square into small rectangles until each rectangle contains approximately t points. Find the optimal RSMT for each rectangle and combine these trees to give the final tree, after some clean-up steps. If each optimal subproblem is solved in $f(t)$ steps then the algorithm runs in $O(f(t)n + n \log n)$ time, which is $O(n \log n)$ for $t = O(\log \log n)$ if, say, Levin's algorithm is used. They show that their approximation is within a factor of $1 + O(1/\sqrt{t})$ of the length of an optimal RSMT, with probability $1 - o(1)$. Another slightly faster algorithm was presented that depends heavily on the uniformity of the point distribution. Hence, even though the same probabilistic bound above holds, this algorithm has a worst-case performance that is at least a factor of t of the length of an optimal RSMT. No implementation was attempted.

Jiang and Tang [99] begin by collecting statistics for all subproblems using 3 vertices from N. Then, after some additional preprocessing, a Kruskal-based algorithm is given that favors Steiner points and edges suggested by the statistics.

Two recent heuristics are presented as automatic wire routing algorithms, for multiterminal nets. Xiong [197] presented an interesting two phase approach. It is based on the underlying grid graph; to find a nearest neighbor a breadth-first search is done. The first phase builds an RMST using Prim's algorithm and tries both embeddings of each edge and marks commonly used grid edges. The second phase is much the same as the first except embeddings are chosen that favor marked commonly used edges; some backtracking is done. Hsu et al. [85] used straightforward versions of Prim's and Kruskal's algorithms but he tried different schemes when a "nearest neighbor" calculation was done. The schemes cleverly take into account the "magnetic attractions" of the other points in N.

4.3. Special Cases

The principle special cases studied restrict the placement of the points in N. Aho, Garey, and Hwang [2] studied the case where all the points lie on the boundary of the grid graph G_N, which has p rows and q columns. They gave an $O(p^2 q + p q^2)$ or $O(n^3)$ time dynamic programming algorithm for this case. It was improved to $O(n)$ time if $p = 2$. Recently Agarwal and Shing [1] have given an $O(p + q)$ time algorithm.

An extension to the above approach is to restrict the points of N to lie on the boundary of their rectilinear convex hull; i.e., each point has one empty open quadrant. Independently Bern [10] and Provan [132] gave $O(n^6)$ time algorithms for this case, based on their algorithms for 1-planar graphs. Bern improved the result to $O(n^5)$ time, by using advanced data structures.

Farley et al. [56] looked at minimal distance rectangle trees (MDRT). Basically, a rectangle tree is a subgraph of G_N where every point is on the outer boundary and is adjacent to two other boundary points. An MDRT is a rectangle tree that spans N and preserves the length of shortest paths between N-points in G_N, They showed that given an instance restricted to an MDRT then the RSMT can be found in linear time. This is a restricted form of the convex case above. Winter [193] gives a polynomial time algorithm for deciding if an MDRT exists and exhibiting one if it does.

4.4. Generalizations

There are few generalizations of the RSMT *per se*. Smith and Gross [155] consider generalizing the metric from the rectilinear, or L_1, metric to the L_p metric, i.e., $(|x_a - x_b|^p + |x_a - x_b|^p)^{1/p}$. They argue that the only interesting range is $1 \le p \le 2$, with respect to applications. They begin with an MST constructed using a generalized Voronoi diagram, and do local steinerizations.

Sankoff and Rousseau's [141] algorithm is for the m-dimensional rectilinear metric; but only for fixed topologies. As mentioned in the previous section, Nastansky et al. [123] and Trubin [172] discussed rooted directed versions of a m-dimensional rectilinear metric. Gilbert and Pollak [75] and Foulds [65] have shown that the upper

bound on L_{RMST} / L_{OPT} goes to ½ as m increases; see also [139].

5. Phylogenetic Trees

The construction of a phylogenetic or evolutionary tree is a basic problem in biological systematics. Such a tree has vertices identified with extant species (operational taxonomic units) and other vertices associated with hypothetical ancestral species. Typically the tree is rooted and the root is regarded as the progenitor. Clearly this is an area where Steiner trees are the appropriate mathematical model, but it is difficult to agree on the point space and what metric should be used.

The literature is, of course, quite large and only relevent pointers into it are given below. Explicit biological motivation is given only rarely in the cited literature [16, 17]. Often the easier problem of finding the phylogenetic tree given its topology is solved, i.e., finding the best positions for the given Steiner points. This is justified since an educated guess of the topology can be made from other biological evidence. The literature can be grouped into three major areas.

The first major area presumes that experts have been able to identify *characters* of the species. Further, each species is labelled with m states corresponding to m different characters. Each character has an associated set of character states, which can be strictly ordered, partially ordered, or unordered. A problem is *cladistic* if they are strictly ordered and *qualitative* if they are unordered. A character with 2 states is *binary*. Species are points in the space of the m-dimensional cross-product of the character states. Suppose that the "distance" between the states of a given character are given. The distance between two points is the "rectilinear" distance, i.e., the sum of the m distances across the various dimensions. Usually character states are identified with integer values and the distance is the rectilinear metric itself. The lowest integer is the ancestral state in the cladistic case; in the "qualitative" case the ancestor is to be chosen but it is cladistic relative to that choice (since the integers imply an ordering). (For additional discussion see [15, 57, 138].)

Four subcases are identified [38], each with its variants . A *Wagner* tree is a Steiner tree where the character states are integers; typically it is unrooted. A *Camin-Sokal* tree is a Steiner tree, where all edges are between points that reflect a change from an ancestral state to a derived state [15, 84]. This is equivalent to the directed rectilinear Steiner problem where all edges are directed away from the origin (progenitor); the algorithms from the previous section can be applied here. There are two subcases: the root is specified or it is chosen. A *Dollo* tree is a Steiner tree, with binary characters, such that no transition between the two states of a character occurs twice in the tree (this assumes $m \geq n-1$). The *chromosone inversion* tree is a Steiner tree with ternary characters with biologically motivated restrictions on the tree edges. All of the problems have been rigorously defined and their decision problems shown to be NP-complete [38, 39, 41, 62, 77]. Perhaps the most interesting algorithmic work has been on the Wagner problem for fixed topologies [57, 141]. Heuristics have been published [15, 57].

The second major area starts with genetic or molecular strings for each of the species. There are three subcases: the Hamming metric [59, 63, 122], the euclidean metric [16], and the minimum mutation metric [17, 144].

The Hamming metric assumes the characters (or nucleotides or alleles or codons) of two sequences can be compared position by position. Hence, formally, the distance between two strings of m characters is the number of positions where characters mismatch. In a *binary* problem each position can be one of two characters. (Note that the binary Hamming metric problem is equivalent to the binary Wagner tree problem, which remains NP-complete [62].) Heuristics have been presented [66] and optimal algorithms for a given topology are known [59, 83, 141, 145]. Note that when the topology is fixed each dimension can be solved independently.

The euclidean metric approach also assumes that sequences can be compared position by position. Cavalli-Sforza and Edwards [16] map each original sequence into a point in m-dimensional euclidean space and give local steinerization heuristics. Thompson [168] continued this work and Rogers [137] gave related results.

The minimal mutation approach assumes, more realistically, that mutations do not just change the value at a position, but that positions are deleted and added. This makes "alignment" difficult and sophisticated definitions of the mutation distance (or "edit distance") between strings are needed. In fact, so much evolutionary change has occurred that it is hard to find sufficiently long subsequences of genetic material to allow comparisons. The work in this area is based on Sankoff's algorithm [142] for fixed topologies. This is remarkable due to the complexity of the metric. It is efficient for long strings but is exponential in n. Various costs can be assigned to the different operations (mutate, delete, insert) depending on their naturally occurring frequencies [143]. Of course, by varying these costs different optimal trees are discovered.

Several algorithms have been used. For a fixed topology there is Sankoff's algorithm [141, 142, 145], as well as an iterative improvement approach [143]. For large n Sankoff's algorithm must be used with iterative improvement. Two approaches have been used to generate topologies. One approach uses an algorithm that does iterative improvement by subtree swapping [17]. Another approach is to generate all topologies satisfying restrictions, usually biological constraints. (To bolster other independent data for imposing restrictions, a biological study [78] used smaller unrestricted runs on troublesome sets of species.)

The third major area in the study of phylogenetic trees is the so called *additive tree* approach. This is a Steiner tree problem only in that new vertices are introduced. A *dissimiliarity matrix* is given that provides lower bounds on the distances between pairs of species, using a common scale. The goal is to construct a tree, possibly with additional nodes, over the n species such that the path distances in the tree for every pair satisfy the given bounds. The tree may be arbitrary or it may be constrained to be a subgraph of a given weighted graph. For complexity results and additional references see [38, 42]. This is considered to be an inferior approach.

Other methods have been used. One method uses a variant of the GSMT using a *lower evaluation* metric, defined in terms of a finite poset [38]. Another, the *cladistic compatibility method*, is not essentially a Steiner problem [40]. Many others have been proposed but they lack sufficient mathematical rigor to be considered here.

6. References

1. P. K. Agarwal and M. T. Shing, Algorithms for the Special Cases of Rectilinear Steiner Trees: I. Points on the Boundary of a Rectilinear Rectangle, Tech. Report TRCS86-17, Univ. of California at Santa Barbara, 1986.
2. A. V. Aho, M. R. Garey and F. K. Hwang, Rectilinear Steiner Trees: Efficient Special-Case Algorithms, *Networks*, **7**, 1977, pp. 37-58.
3. Y. P. Aneja, An Integer Linear Programming Approach to the Steiner Problem in Graphs, *Networks*, **10**, 1980, pp. 167-178.
4. A. Balakrishnan, Formulations and Algorithms for the Steiner Network Problem, Unpublished manuscript, Sloan School Manag., MIT, 1982.
5. A. Balakrishnan and N. R. Patel, Problem Reduction Methods and a Tree Generation Algorithm for the Steiner Network Problem, *Networks*, **17**, 1987, pp. 65-85.
6. J. E. Beasley, An Algorithm for the Steiner Problem in Graphs, *Networks*, **14**, 1984, pp. 147-159.
7. J. E. Beasley, An SST-Based Algorithm for the Steiner Problem on Graphs, Working Paper, Dept Manag. Sci., Imperial College, London, 1985.
8. M. W. Bern and M. de Carvalho, A Greedy Heuristic for the Rectilinear Steiner Tree Problem, Tech. Report, Computer Science Division, Univ. of California at Berkeley, 1985.
9. M. W. Bern, Two Probabilistic Results on Rectilinear Steiner Trees, *Algorithmica*, to appear.
10. M. W. Bern, Network Design Problems: Steiner Trees and Spanning k-Trees, Ph.D. Dissertation, Computer Science Division, Univ. of California at Berkeley, 1987.
11. S. Bhaskaran, Optimal Design of Gas Pipeline Networks, Ph.D. dissertation, Univ. Adelaide, Australia, 1978.
12. S. Bhaskaran and F. J. M. Salzborn, Optimal Design of Gas Pipeline Networks, *J. Operational Research Society*, **30**, 1979, pp. 1047-1060.
13. W. M. Boyce and J. B. Seery, STEINER72: An Improved Version of the Minimal Network Problem, Tech Rept 35, Comp. Sci. Res. Ctr., Bell Labs, 1975.
14. W. M. Boyce, An Improved Program for the Full Steiner Tree Problem, *ACM Trans. Math. Software*, **3**, 1977, pp. 339-385.
15. J. H. Camin and R. R. Sokal, A Method for Deducing Branching Sequences in Phylogeny, *Evolution*, **19**, 1965, pp. 311-326.
16. L. L. Cavalli-Sforza and A. W. F. Edwards, Phylogenetic Analysis: Models and Estimation Procedures, *Am. J. Human Genetics*, **19**, 1967, pp. 233-257.
17. R. J. Cedergren, D. Sankoff, B. LaRue and H. Grosjean, The Evolving tRNA Molecule, *Critical Reviews in Biochemistry*, **11**, June 1981, pp. 35-104.
18. S. K. Chang, The Generation of Minimal Trees With a Steiner Topology, *J. ACM*, **19**, 1972, pp. 699-711.
19. N. Chen, New Algorithm for Steiner Tree on Graphs, *IEEE Symp on Circuits and Systems*, 1983, pp. 1217-1219.
20. T. Cheung and J. Soukup, A Note on Minimum Cost Networks with Nonlinear Costs, *SIAM J. Appl. Math*, **34**, 1978, pp. 225-226.
21. F. R. K. Chung and E. N. Gilbert, Steiner Trees for the Regular Simplex, *Bull. Inst. Math. Acad. Sinica*, **4**, 1976, pp. 313-325.
22. F. R. K. Chung and R. L. Graham, Steiner Trees for Ladders, *Ann. Disc. Math.*, **2**, 1978, pp. 173-200.
23. F. R. K. Chung and F. K. Hwang, A Lower Bound for the Steiner Tree Problem, *SIAM J. Appl. Math*, **34**, 1978, pp. 27-36.
24. F. R. K. Chung and F. K. Hwang, The Largest Minimal Rectilinear Steiner Trees for a Set of n Points Enclosed in a Rectangle with Given Perimeter, *Networks*, **9**, 1979, pp. 19-36.
25. F. R. K. Chung and R. L. Graham, On Steiner Trees for Bounded Point Sets, *Geometriae Dedicata*, **11**, 1981, pp. 353-361.

26. F. R. K. Chung and R. L. Graham, A New Bound for Euclidean Steiner Minimal Trees, *Ann. N.Y. Acad. Sci.*, **440**, 1985, pp. 328-346.

27. F. R. K. Chung, R. L. Graham and M. E. Saks, Dynamic Search in Graphs, in *Discrete Algorithms and Complexity*, D. S. J. al. (ed.), 1987, 351-387.

28. R. C. Clark, Communication Networks, Soap Films, and Vectors, *Phys. Educ.*, **16**, 1981, pp. 32-37.

29. E. J. Cockayne and Z. A. Melzak, Steiner's Problem for Set-Terminals, *Quart. Appl. Math.*, **26**, 1967, pp. 213-218.

30. E. J. Cockayne, On the Steiner Problem, *Canad. J. Math.*, **10**, 1967, pp. 431-450.

31. E. J. Cockayne and Z. A. Melzak, Euclidean Contructability in Graph-Minimization Problems, *Math Magazine*, **42**, 1969, pp. 206-208.

32. E. J. Cockayne, On the Efficiency of the Algorithm for Steiner Minimal Trees, *SIAM J. Appl. Math.*, **18**, 1970, pp. 150-159.

33. E. J. Cockayne and D. G. Schiller, Computation of a Steiner Minimal Tree, in *Combinatorics*, D. J. A. W. D. R. Woodall (ed.), Inst. Math. Appl., 1972, 53-71.

34. E. J. Cockayne and D. B. Peters, MEST82, A Program for Minimal Full Steiner Trees in the Plane, *Congressus Numeratium (Proc. 14th S. E. Conf. on Combinatorics, Graph Theory and Computing)*, **39**, 1983, pp. 203-224.

35. E. J. Cockayne and D. E. Hewgill, Exact Computation of Steiner Minimal Trees in the Plane, *Inform. Proc. Letters*, **22**, 1986, pp. 151-156.

36. G. Cornuejols, J. Fonlupt and D. Naddef, The Graphical Travelling Salesman Problem and Some Related Integer Polyhedra, Research Rept. 378, Appliquees de Grenoble, 1983.

37. R. Courant and H. Robbins, *What is Mathematics?*, Oxford Univ. Press, New York, 1941.

38. W. H. E. Day, Computationally Difficult Parsimony Problems in Phylogenetic Systematics, *J. Theor. Biology*, **103**, 1983, pp. 429-438.

39. W. H. E. Day, D. S. Johnson and D. Sankoff, The Computational Complexity of Inferring Rooted Phylogenies, *Math. Biosciences*, **81**, 1986, pp. 33-42.

40. W. H. E. Day and D. Sankoff, Computational Complexity of Inferring Phylogenies by Compatibility, *Systematic Zoology*, **35**, 1986, pp. 224-229.

41. W. H. E. Day and D. Sankoff, Computational Complexity of Inferring Phylogenies from Chromosone Inversion Data, *J. Theor. Biology*, **124**, 1987, pp. 213-218.

42. W. H. E. Day, Computational Complexity of Inferring Phylogenies from Dissimilarity Matrices, *Bull. Math. Biology*, to appear.

43. S. E. Dreyfus and R. A. Wagner, The Steiner Problem in Graphs, *Networks*, **1**, 1972, pp. 195-207.

44. D. Z. Du, E. Y. Yao and F. K. Hwang, A Short Proof of a Result of Pollak on Steiner Minimal Trees, *J. Combinatorial Theory. Ser. A*, **32**, 1982, pp. 396-400.

45. D. Z. Du and F. K. Hwang, A New Bound for the Steiner Ratio, *Trans. Amer. Math. Soc*, **278**, 1983, pp. 137-148.

46. D. Z. Du, F. K. Hwang and J. F. Weng, Steiner Minimal Trees on Zig-Zag Lines, *Trans. Amer. Math. Soc.*, **278**, 1983, pp. 149-156.

47. D. Z. Du, F. K. Hwang and E. N. Yao, The Steiner Ratio Conjecture is True for Five Points, *J. Combinatorial Theory. Ser. A*, **38**, 1985, pp. 230-240.

48. D. Z. Du, F. K. Hwang and S. C. Chao, Steiner Minimal Tree for Points on a Circle, *Proc. Amer. Math. Soc.*, **95**, 1985, pp. 613-618.

49. D. Z. Du and F. K. Hwang, On a Conjecture of Trietsch and Handler on the Flow-Dependent Steiner Ratio, *Networks*, **16**, 1986, pp. 47-50.

50. D. Z. Du, F. K. Hwang and J. F. Weng, Steiner Minimal Trees for Regular Polygons, *Disc. and Comput. Geom.*, **2**, 1987, pp. 65-87.

51. D. Z. Du, F. K. Hwang, G. D. Song and G. Y. Ting, Steiner Minimal Trees on Sets of Four Points, *Disc. and Comput. Geom.*

52. D. Z. Du and F. K. Hwang, Steiner Minimal Trees for Bar Waves, *J. Appl. Math.*.

53. C. W. Duin and A. Volgenant, Some Generalizations of the Steiner Problem in Graphs, *Networks*, **17**, 1987, pp. 353-364.

54. C. El-Arbi, Une Heuristique pour le Probleme de l'Arbre de Steiner (in French), *RAIRO (Operations Research)*, **12**, 1978, pp. 207-212.

55. R. E. Erickson, C. L. Monma and A. F. Veinott, Send-and-Split Method for Minimum-Concave-Cost Network Flows, *Mathematics of Operation Research.*

56. A. M. Farley, S. T. Hedetniemi and S. L. Mitchell, Rectilinear Steiner Trees in Rectangle Trees, *SIAM J Alg Disc Meth*, **1**, 1980, pp. 70-81.

57. J. S. Farris, Methods for Computing Wagner Trees, *Systematic Zoology*, **19**, 1970, pp. 83-92.

58. L. Few, The Shortest Path and Shortest Road Through n Points, *Mathematika*, **2**, 1955, pp. 141-144.

59. W. M. Fitch, Towards Defining a Course of Evolution: Minimum Change for a Specific Tree Topology, *Systematic Zoology*, **20**, 1971, pp. 406-416.

60. L. R. Foulds and P. B. Gibbons, A Branch and Bound Approach to the Steiner Problem in Graphs, *Proc 14th Ann Conf Operational Research Soc of New Zealand*, **1**, 1978, pp. 61-70.

61. L. R. Foulds, P. B. Gibbons and M. L. Shore, Algorithms for the Steiner Problem in Graphs, *J. Combin. Inf. Syst. Sci.*, **6**, 1981, pp. 215-219.

62. L. R. Foulds and R. L. Graham, The Steiner Problem in Phylogeny is NP-Complete, *Advances in Applied Math.*, **3**, 1982, pp. 43-49.

63. L. R. Foulds, Testing the Theory of Evolution, *Optima*, **10**, 1983, pp. 1-3.

64. L. R. Foulds and V. J. Rayward-Smith, Steiner Problems in Graphs: Algorithms and Applications, *Engineering Optimization*, **7**, 1983, pp. 7-16.

65. L. R. Foulds, Maximum Savings in the Steiner Problem in Phylogeny, *J. Theor. Biology*, **107**, 1984, pp. 471-474.

66. L. R. Foulds, Testing the Theory of Evolution: A Novel Application of Combinatorial Optimization, *Discrete Appl. Math*, **15**, 1986, pp. 271-282.

67. Y. Fu, Application of Linear Graph Theory to Printed Circuits, *Proc. Asilomar Conf. Systems and Circuits*, 1967, pp. 721-728.

68. R. L. Gallawa, Application of the Steiner Street Problem to a More Economical Data Bus, *Fiber and Integrated Optics*, **1**, 1978, pp. 289-311.

69. M. R. Garey and D. S. Johnson, The Rectilinear Steiner Tree Problem is NP-Complete, *SIAM J. Appl. Math.*, **32**, 1977, pp. 826-834.

70. M. R. Garey, L. R. Graham and D. S. Johnson, The Complexity of Computing Steiner Minimal Trees, *SIAM J. Appl. Math.*, **32**, 1977, pp. 835-859.

71. M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.

72. G. Georgakopoulos and C. H. Papdimitriou, A 1-Steiner Tree Problem, *J Algorithms*, **8**, 1987, pp. 122-130.

73. E. N. Gilbert, Random Minimal Trees, *Journal SIAM*, **13**, 1965, pp. 376-387.

74. E. N. Gilbert, A Minimum Cost Communication Networks, *Bell System Tech. J.*, **46**, 1967, pp. 2209-2227.

75. E. N. Gilbert and H. O. Pollak, Steiner Minimal Trees, *SIAM J. Appl. Math*, **16**, 1968, pp. 1-29.

76. R. L. Graham and F. K. Hwang, Remarks on Steiner Minimal Trees I, *Bull. Inst. Math. Acad. Sinica*, **4**, 1976, pp. 177-182.

77. R. L. Graham and L. R. Foulds, Unlikelihood that Minimal Phylogenies for a Realistic Biological Study Can Be Constructed in Reasonable Computation Time, *Math Biosciences*, **60**, 1982, pp. 133-142.

78. M. W. Gray, D. Sankoff and R. J. Cedergren, On the Evolutionary Descent of Organisms and Organelles: A Global Phylogeny Based on a Highly Conserved Structural Core in Small Subunit Ribosomal RNA, *Nucleic Acids Research*, **12**, 1984, pp. 5837-5852.

79. S. L. Hakimi, Steiner's Problem in Graphs and Its Implications, *Networks*, **1**, 1971, pp. 113-133.

80. M. Hanan, Net Wiring for Large Scale Integrated Circuits, RC 1375, IBM Research Report, 1965.

81. M. Hanan, On Steiner's Problem with Rectilinear Distance, *Journal SIAM Applied Math.*, **14**, 1966, pp. 255-265.

82. M. Hanan, A Counterexample to a Theorem of Fu on Steiner's Problem, *IEEE Trans. on Circuit Theory*, **CT-19**, 1972, pp. 74.

83. J. A. Hartigan, Minimum Mutations Fits to a Given Tree, *Biometrics*, **29**, 1973, pp. 53-65.

84. J. A. Hendrickson, Clustering in Numerical Cladistics: A Minimum-Length Directed Tree Problem, *Math. Biosciences*, **3**, 1968, pp. 371-381.

85. Y. C. Hsu, Y. Pan and W. J. Kubitz, A Path Selection Global Router, *24th ACM/IEEE Design Automation Conference*, 1987, pp. 641-644.

86. F. K. Hwang, On Steiner Minimal Trees with Rectilinear Distance, *SIAM Journal Applied Math.*, **30**, 1976, pp. 104-114.

87. F. K. Hwang, The Rectilinear Steiner Problem, *Design Automation & Fault Tolerant Computing*, **3**, 1978, pp. 303-310.

88. F. K. Hwang, An O(n log n) Algorithm for Suboptimal Rectilinear Steiner Trees, *IEEE Trans. on Circuits and Systems*, **CAS-26**, 1979, pp. 75-77.

89. F. K. Hwang, An O(n log n) Algorithm for Rectilinear Minimal Spanning Trees, *Journal ACM*, **26**, 1979, pp. 177-182.

90. F. K. Hwang, J. F. Weng and D. Z. Du, A Class of Full Steiner Minimal Trees, *Disc. Math.*, **45**, 1983, pp. 107-112.

91. F. K. Hwang, A Linear Time Algorithm for Full Steiner Trees, *Oper. Res. Letters*, **5**, 1986, pp. 235-237.

92. F. K. Hwang and J. F. Weng, Hexagonal Coordinate Systems and Steiner Minimal Trees, *Disc. Math*, **62**, 1986, pp. 49-57.

93. F. K. Hwang, G. D. Song, G. Y. Ting and D. Z. Du, A Decomposition Theorem for Steiner Minimal Trees, *Disc. and Comput. Geom.*.

94. A. Iwainsky, Some Notes on the Steiner Tree Problem in Graphs, in *Optimization of Connection Structures in Graphs*, A. Iwainsky (ed.), Centr. Inst. of Cyb. a. Inform. Processes, Berlin, 1985, 57-73.

95. A. Iwainsky, E. Canuto, O. Taraszow and A. Villa, Network Decomposition for the Optimization of Connection Structures, *Networks*, **16**, 1986, pp. 205-235.

96. A. Iwainsky and A. Nuemann, Some Variants of a Heuristic Approach to the Solution of the Steiner Tree Problem in Graphs, , 1986.

97. A. Jain, Probabilistic Analysis of LP Relaxation Bounds for the Steiner Problem in Networks, Working Paper, The Wharton School, Univ. Penn., 1987.

98. V. Jarnik and O. Kossler, O mimimálnich gratech obbsahujicich n daných bodu (in Czech), *Casopis Pesk. Mat. Fyr.*, **63**, 1934, pp. 223-235.

99. J. W. Jiang and P. S. Tang, An Algorithm for Generating Rectilinear Steiner Trees [in Chinese], *J. of Fudan Univ. (Natural Science)*, **25**, 1986, pp. 343-349.

100. R. R. Kallman, A Conjecture of Gilbert and Pollak on Minimal Trees, *Studies in Appl. Math.*, **22**, 1973, pp. 141-151.

101. R. M. Karp, Reducibility Among Combinatorial Problems, in *Complexity of Computer Computations*, R. E. M. J. W. Thatcher (ed.), Plenum Press, New York, 1972, 85-103.

102. R. M. Karp, The Probabilistic Analysis of Some Combinatorial Search Algorithms, in *Algorithms and Complexity: New Directions and Recent Results*, J. Traub (ed.), Academic Press, New York, 1976, 1-19.

103. I. N. Katz and L. Cooper, Facility Location in the Presence of Forbidden Regions, I, *European J. of Operational Research*, **6**, 1981, pp. 166-173.

104. J. Komlos and M. T. Shing, Probabilistic Partitioning Algorithms for the Rectilinear Steiner Problem, *Networks*, **15**, 1985, pp. 413-423.

105. P. Korhonen, An Algorithm For Transforming a Spanning Tree into a Steiner Tree, *Survey of Math programming (Proc 9th Intl Math Prog Symp)*, **2**, 1979, pp. 349-357, North-Holland.

106. A. Kotzig, Optimálne spojovacie systemy (in Czech), in *Mathematické Metódy v Hospodársky Praxi*, V. Kapitoly (ed.), Vydavatelstvo Slovenskej Akadémie Vied, Bratislava, 1961.

107. L. Kou, G. Markowsky and L. Berman, A Fast Algorithm for Steiner Trees, *Acta Informatica*, **15**, 1981, pp. 141-145.

108. L. Kucera, A. Marchetti-Spaccamela, M. Protasi and M. Talamo, Near Optimal Algorithms for Finding Minimum Steiner Trees on Random Graphs, in *Mathematical Foundations of Computer Science 1986 (LNCS 233)*, J. G. al. (ed.), Springer-Verlag, 1986, 501-511.

109. H. W. Kuhn, " Steiner's" Problem Revisited, in *Studies in Optimization*, vol. 10, G. B. D. B. C. Eaves (ed.), Math Assoc. of America, 1975, 98-107.

110. D. H. Lee, Low Cost Drainage Networks, *Networks*, **6**, 1976, pp. 351-371.

111. J. H. Lee, N. K. Bose and F. K. Hwang, Use of Steiner's Problem in Supoptimal Routing in Rectilinear Metric, *IEEE Trans. on Circuits and Systems*, **CAS-23**, 1976, pp. 470-476.

112. A. J. Levin, Algorithm for the Shortest Connection of a Group of Graph Vertices, *Soviet Math. Doklady*, **12**, 1971, pp. 1477-1481.

113. A. Liestman, Construction of Steiner Trees with Obstacles in the Plane, Masters Thesis, Dept Comp Science, Univ Illinois , 1978.

114. M. Lundy, Applications of the Annealing Algorithm to Combinatorial Problems in Statistics, *Biometrika*, **72**, 1985, pp. 191-198.

115. M. Lundy and A. Mees, Convergence of an Annealing Algoithm, *Math. Prog.*, **34**, 1986, pp. 111-124.

116. S. Ma, The Steiner Tree Problem on Graph and Its Heuristic Algorithm (in Chinese), *Chinese J Computing*, **8**, 1985, pp. 237-239.

117. N. Megiddo, Cost Allocation for Steiner Trees, *Networks*, **8**, 1978, pp. 1-6.

118. Z. A. Melzak, On the Problem of Steiner, *Canad. Math. Bull.*, **4**, 1961, pp. 143-148.

119. Z. A. Melzak, *Companion to Concrete Mathematics*, Wiley, New York, 1973.

120. Z. A. Melzak and J. S. Forsyth, Polyconics 1: Polyellipses and Optimization, *Quart. J. Math.*, 1977, pp. 239-255.

121. W. Miehle, Link-length Minimization in Networks, *Oper. Res.*, **6**, 1958, pp. 232-243.

122. G. W. Moore, J. Barnabas and M. Goodman, A Method for Constructing Maximum Parsimony Ancestral Amino Acid Sequences on a Given Network, *J. Theor. Biol*, **38**, 1973, pp. 459-485.

123. L. Nastansky, S. M. Selkow and N. F. Stewart, Cost-Minimal Trees in Directed Acyclic Graphs, *Zeitschrift fur Operations Research*, **18**, 1974, pp. 59-67.

124. A. P. Ng, P. Raghavan and C. D. Thompson, A Language for Describing Rectilinear Steiner Tree Configurations, *23rd ACM/IEEE Design Automation Conf*, 1986, pp. 659-662.

125. K. Ollerenshaw, Minimum Networks Linking Four Points in a Plane, *Inst. Math. Appl.*, **15**, 1978, pp. 208-211.

126. F. P. Palermo, A Network Minimzation Problem, *IBM J. Res. and Dev.*, **5**, 1961, pp. 335-337.

127. J. Plesnik, A Bound for the Steiner Tree Problem in Graphs, *Math Slovaca*, **31**, 1981, pp. 155-163.

128. H. O. Pollak, Some Remarks on the Steiner Problem, *J. Combinatorial Theory. Ser. A*, **24**, 1978, pp. 278-295.

129. G. Polya, *Induction and Analogy in Mathematics*, Princeton Univ. Press, New Jersey, 1954.

130. A. Prodon, T. M. Liebling and H. Groflin, Steiner's Problem on Two-Trees, Tech Rept RO850315, Ecole Polytechnique Federale de Lausanne, 1985.

131. J. S. Provan, An Approximation Scheme for Finding Steiner Trees with Obstacles, Tech Rept UNC/OR/Tech. Rep.-87/9, Dept Operations Research, Univ N. Carolina, Chapel Hill, 1987.

132. J. S. Provan, Convexity and the Steiner Tree Problem, *Networks*.

133. R. L. Rardin, R. G. Parker and M. B. Richey, A Polynomial Algorithm for a Class of Steiner Tree Problems on Graphs, Indus. and Syst. Engr. Rept. Series J-82-5, Georgia Inst. Tech., 1982.

134. V. J. Rayward-Smith, The Computation of Nearly Minimal Steiner Trees in Graphs, *Intl. Journal Math. Educ. Sci. Technol.*, **14**, 1983, pp. 15-23.

135. V. J. Rayward-Smith and A. Clare, On Finding Steiner Vertices, *Networks*, **16**, 1986, pp. 283-294.

136. D. Richards, Fast Heuristic Algorithms for Rectilinear Steiner Trees, *Algorithmica*.

137. J. S. Rogers, Deriving Phylogenetic Trees From Allele Frequencies, *Systematic Zoology*, **33**, 1984, pp. 52-63.

138. F. J. Rohlf, A Note on Minmum Length Trees, *Systematic Zoology*, **33**, 1984, pp. 341-343.

139. M. H. Rudowski, Hypothesis on the Length of Minimal Steiner Tree in Multidimensional Spaces With a Rectilinear Metric (in Polish), *Arch. Autom. & Telemech*, **29**, 1984, pp. 353-358.

140. T. M. Sandifer, Algorithms for Solving the Directed Minimal Steiner Tree Problem, Ph.D. dissertation, Univ. Mass, Amherst, 1981.

141. D. Sankoff and P. Rousseau, Locating the Vertices of a Steiner Tree in an Arbitrary Metric Space, *Math Programming*, **9**, 1975, pp. 240-246.

142. D. Sankoff, Minimal Mutation Trees of Sequences, *SIAM J Appl Math*, **28**, 1975, pp. 35-42.

143. D. Sankoff, R. J. Cedergren and G. Lapalme, Frequency of Insertion-Deletion, Transversion, and Transition in the Evolution of 5S Ribosomal RNA, *J. Molecular Evolution*, **7**, 1976, pp. 133-149.

144. D. Sankoff, R. J. Cedergren and W. McKay, A Strategy for Sequence Phylogeny Research, *Nucleic Acids Research*, **10**, 1982, pp. 421-431.

145. D. Sankoff and R. J. Cedergren, Simultaneous Comparison of Three or More Sequences Related by a Tree, in *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, D. S. J. B. Kruskal (ed.), Addison-Wesley, 1983, 253-263.

146. A. Segev, The Node-Weighted Steiner Tree Problem, *Networks*, **17**, 1987, pp. 1-17.

147. M. Servit, Heuristic Algorithms for Rectilinear Steiner Trees, *Digital Processes*, **7**, 1981, pp. 21-32.

148. M. I. Shamos and D. Hoey, Closest-Point Problems, *16th Annual Symp. on Foundations of Computer Science*, 1975, pp. 151-162.

149. M. L. Shore, L. R. Foulds and P. B. Gibbons, An Algorithm for the Steiner Problem in Graphs, *Networks*, **12**, 1982, pp. 323-333.

150. J. M. Smith, Generalized Steiner Network Problems, Ph.D. dissertation, Univ. Illinois, Urbana, 1978.

151. J. M. Smith and J. S. Liebman, Steiner Trees, Steiner Circuits and the Interference Problem in Building Design, *Engineering Optimization*, **4**, 1979, pp. 15-36.

152. J. M. Smith, D. T. Lee and J. S. Liebman, An O(N log N) Heuristic Algorithm for the Rectilinear Steiner Minimal Tree Problem, *Engineering Optimization*, **4**, 1980, pp. 179-192.

153. J. M. Smith and J. S. Liebman, An $O(n^2)$ Heuristic Algorithm for the Directed Steiner Minimal Tree Problem, *Appl. Math. Modelling*, **4**, 1980, pp. 369-375.

154. J. M. Smith, D. T. Lee and J. S. Liebman, An O(N log N) Heuristic for Steiner Minimal Tree Problems on the Euclidean Metric, *Networks*, **11**, 1981, pp. 23-29.

155. J. M. Smith and M. Gross, Steiner Minimal Trees and Urban Service Networks, *J. Socio. Econ. Plng.*, **16**, 1982, pp. 21-38.

156. J. M. Smith, Steiner Minimal Trees with Obstacles, Tech Rept, Dept Indust. Engr. and Oper. Res., Univ Mass., 1982.

157. J. Soukup and W. F. Chow, Set of Test Problems of the Minimum Length Connection Networks, *SIGMAP Newsletter*, **15**, 1973, pp. 48-51.

158. J. Soukup, On Minimum Cost Networks With Non-Linear Costs, *SIAM J. Appl. Math.*, **29**, 1975, pp. 571-581.

159. J. Soukup, Minimum Steiner Trees, Roots of a Polynomial and Other Magic, *ACM/SIGMAP Newsletter*, **22**, 1977, pp. 37-51.

160. H. Steinhaus, *Mathematical Snapshots*, Oxford Univ. Press, New York, 1960.

161. G. F. Sullivan, Approximation Algorithms for Steiner Tree Problems, Tech Rept 249, Dept Computer Science, Yale Univ, 1982.

162. J. W. Suurballe, Algorithms for Minimal Trees and Semi-Steiner Trees Based on the Simplex Method (Abstract), *Proc Princeton Symp on Math Programming*, 1970, pp. 614-615.

163. H. Takahashi and A. Matsuyama, An Approximate Solution for the Steiner Problem in Graphs, *Math. Japonica*, **24**, 1980, pp. 573-577.

164. K. Takamizawa, T. Nishizeki and N. Saito, Linear-Time Computability of Combinatorial Problems on Series-Parallel Graphs, *J. ACM*, **29**, 1982, pp. 623-641.

165. O. Taraszow and A. Iwainsky, Trassierung und Plazierung auf Graphenmodellen, *Vortrags-Sammelband der 4. Fachtagung Numerische Realisierung Math. Modelle*, 1981, pp. 215-231.

166. O. Taraszow and P. Richter, New Heuristic Algorithms for Solving the Steiner Tree Problem in Graphs, , 1986.

167. C. D. Thomborson (a.k.a. Thompson), L. L. Deneen and G. M. Shute, Computing a Rectilinear Steiner Minimal Tree in $n^{O(\sqrt{n})}$ Time, in *Parallel Algorithms and Architectures*, A. al. (ed.), Akademie-Verlag, Berlin, 1987, 176-183.

168. E. A. Thompson, The Method of Minimum Evolution, *Ann. Human Genetics, London*, **36**, 1973, pp. 333-340.

169. M. Tideman, Comment on "A Network Minimization Problem", *IBM J. Res. and Dev.*, **6**, 1962, pp. 259.

170. D. Trietsch and G. Y. Handler, The Gilbert and Pollak Conjecture — A Generalization, *Networks*, **15**, 1983, pp. 365-380.

171. D. Trietsch, Augmenting Euclidean Networks - The Steiner Case, *SIAM J. Appl. MAth.*, **45**, 1985, pp. 855-860.

172. V. A. Trubin, Subclass of the Steiner Problems on a Plane with Rectilinear Metric, *Cybernetics*, **21**, 1985, pp. 320-324.

173. L. P. P. P. van Ginneken and R. H. J. M. Otten, Global Wiring for Custom Layout Design, *Proc of Intl Symp Circuits and Systems*, 1985, pp. 207-208.

174. P. A. van de Heyden, Some Experiments With Steiner Trees, *Internat. Ser. Numer. Math.*, **36**, 1977, pp. 79-109.

175. J. A. Wald and C. J. Colbourn, Steiner Trees in Outerplanar Graphs, *Proc. 13th S.E. Conf. on Combinatorics, Graph Theory, and Computing*, 1982, pp. 15-22.

176. J. A. Wald and C. J. Colbourn, Steiner Trees in Probabilistic Networks, *Microelectron. Reliab.*, **23**, 1983, pp. 837-840.

177. J. A. Wald and C. J. Colbourn, Steiner Trees, Partial 2-Trees, and Minimal IFI Networks, *Networks*, **13**, 1983, pp. 159-167.

178. J. A. Wald and P. G. Sorenson, Resolving the Query Inference Problem Using Steiner Trees, *ACM Trans. Database Sytems*, **9**, 1984, pp. 348-368.

179. J. A. Wald, A Linear Algorithm for a Class of Problems on Partial 2-Trees, Manuscript, Schlumberger Tech. Corp., 1987.

180. S. Wang, A Multiple Source Algorithm for Suboptimum Steiner Trees in Graphs, *Proc W G '85*, 1985, pp. 387-396.

181. J. F. Weng, Steiner Minimal Trees on Vertices of Regular Polygons, *Acta Math. Appl. Sinica*, **8**, 1983, pp. 129-141.

182. J. F. Weng, Generalized Steiner Problems and Hexagonal Coordinate System (in Chinese), *Acta Math. Appl. Sinica*, **8**, 1985, pp. 383-397.

183. C. Werner, Networks of Minimal Length, *The Canad. Geographer*, **13**, 1969, pp. 47-69.

184. K. White, M. Farber and W. Pulleybank, Steiner Trees, Connected Domination and Strongly Chordal Graphs, *Networks*, **15**, 1985, pp. 109-124.

185. Widmayer, Fast Approximation Algorithms for Steiner's Problem in Graphs, Tech. Rept., Inst. fur Angew. Info., Univ. Karlsruhe, 1985.

186. P. Widmayer, On Approximation Algorithms for Steiner's Problem in Graphs, in *Graph-Theoretic Concepts in Computer Science (LNCS 246)*, G. T. G. Schmidt (ed.), Springer-Verlag, 1986, 17-28.

187. P. Winter, The Steiner Problem, M.Sc. Thesis, Inst. of Datalogy, Univ of Copenhagen, Denmark, 1981.

188. P. Winter, An Algorithm for the Steiner Problem in the Euclidean Plane, *Networks*, **15**, 1985, pp. 323-345.

189. P. Winter, Generalized Steiner Problem in Outerplanar Networks, *BIT*, **25**, 1985, pp. 485-496.

190. P. Winter, Generalized Steiner Problem in Series-Parallel Networks, *J. Algorithms*, **7**, 1986, pp. 549-566.

191. P. Winter, Steiner Problem in Networks: A Survey, *Networks*, **17**, 1987, pp. 129-167.

192. P. Winter, Generalized Steiner Problem in Halin Networks, , to appear.

193. P. Winter, Construction of Minimum Distance Rectangle Trees, , to appear.

194. P. Winter, Steiner Problem in Halin Networks, *Disc. Appl. Math.*.

195. R. T. Wong, A Dual Ascent Approach for Steiner Tree Problems on a Directed Graphs, *Math Programming*, **28**, 1984, pp. 271-287.

196. Y. F. Wu, P. Widmayer and C. K. Wong, A Faster Approximation Algorithm for the Steiner Problem in Graphs, *Acta Informatica*, **23**, 1986, pp. 223-229.

197. J. G. Xiong, Algorithms for Global Routing, *23rd ACM/IEEE Design Automation Conference*, 1986, pp. 824-830159-167.

198. Y. Y. Yang and O. Wing, An Algorithm for the Wiring Problem, *Digest of the IEEE Intl Symp on Electrical Networks*, 1971, pp. 14-15.

199. Y. Y. Yang and O. Wing, Supoptimal Algorithm for a Wire Routing Problem, *IEEE Trans. on Circuit Theory*, **CT-19**, 1972, pp. 508-510.

200. Y. Y. Yang and O. Wing, Optimal and Suboptimal Solution Algorithms for the Wiring Problem, *Proc. IEEE Intl. Symp. Circuit Theory*, 1972, pp. 154-158.

201. Y. Y. Yang and O. Wing, On a Multinet Wiring Problem, *IEEE Trans. on Circuit Theory*, CT-20, 1973, pp. 250-252.