Discussion Paper No. 640

CONNECTING EUCLIDEAN NETWORKS--
THE STEINER CASE

by

Dan Trietsch*

January 1985

*Visiting Assistant Professor Operations Management and Decision Sciences, Department of Managerial Economics and Decision Sciences, J. L. Kellogg Graduate School of Management, Northwestern University, Evanston, Illinois 60201.

"Connecting Euclidean Networks--The Steiner Case"

Abstract

On a Euclidean plane n networks are given. It is required to interconnect the networks by links of minimal total length. The use of Steiner points is allowed, and connections can be made anywhere along the edges or to the vertices of the networks. We prove that the problem can be solved in finite time by methods similar to those used for the Euclidean Steiner tree problem. The problem can be generalized to include flow dependent costs for the various links. However, even in the form discussed here, it may be useful for computers, telecommunication or electricty networks connection, to mention a few examples, if the projected internetworks' flow does not justify more than the minimal possible investment.

Keywords: Steiner trees, network design

1. Introduction

The need to connect existing networks is encountered more and more. Prominent examples are the fast abounding computer networks to which we may wish to add new computers or even other networks; communication networks where the international network actually connects local ones, etc.; sewage disposal networks of several cities may have to be connected jointly to a common treatment plant; highway networks such as those belonging to separate cities may have to be connected, or we may want to add connections to some existing ones; several electrical utilities may wish to construct a connecting network so that they can support each other during respective peak hours which do not coincide exactly, etc., etc. In many of the examples mentioned, extra junctions are relatively inexpensive (this is especially true for networks spanning great distances). For instance, in computer networks, the cost of junctions (i.e., multiplexors and so forth) drops a lot faster over the years than does the cost of lines, and the cost of the lines thus becomes more and more dominant. In large water networks, junctions were always cheap, relatively. This leads us to (i) allow connections anywhere along the existing arcs, and (ii) allow the use of Steiner points (i.e., extra junctions). In addition, if we assume that the Euclidean distance is a good approximation to the cost of an arc--thus neglecting (in this paper) the fact that these costs are functions of the flow--we get what we call the Euclidean networks Steiner connection problem, the subject of this paper.

If we consider single nodes (points) as "degenerate" networks, we can see that the Euclidean Steiner tree problem [6, 4, 1, 9] is a special case of our problem. If exactly one of the networks is not degenerate, we have an instance of the network augmenting problem [8], described in some detail below.

It is interesting to compare our problem to other network design problems of similar nature, and see in what sense it is new and challenging. If we allow connections along the arcs, but no Steiner points (a strange policy, this one), then we have an instance of the "regular" minimal spanning tree, a very tractable problem indeed [7]. It is the Steiner points which cause difficulties, which could be expected since they suffice to make the regular minimal spanning tree problem NP-hard [2]. However, it is not trivial to show to what extent the techniques of the Steiner tree problem apply here. Indeed, it seems that some nice properties of Steiner trees which allow some relative efficiency in its solution (see [9]) cannot be generalized, even to a convex version of our problem (i.e., a counterexample exists for the only obvious generalization).[1] However, and this is the main result of this paper, we are able to show that there is a finite procedure for the solution, using the well-known Steiner construction, and guaranteed to converge to a global minimum. It follows that the problem is NP-hard. Actually, in its complexity, the procedure we suggest is comparable to a well-known algorithm for Steiner trees [1].

Before we discuss our case, we mention some results of [8] in detail. Two main results are obtained there: (i) an extension of the Steiner construction which we also need here, where we represent the nodes by one point and connect to the network in a locally optimal manner; and (ii) the Steiner polygon concept, defined by Cockayne [1], is generalized to the case of connecting n points to a segment. In the remainder of the introduction we

---

[1] It seems that the results of [9] can be generalized for a "single segment network" augmenting problem, but this is as far as it goes, and it is not far enough.

briefly describe those results of [8] which we need for this paper. Then we present new results in sections 2 and 3, and conclude with some examples in section 4.

The Network Augmenting Problem

On a Euclidean plane, let a set N of $n \geqslant 1$ points and a network G(V,A) be given, where V is a set of vertices and A is a set of straight edges which span the verices of V. It is required to connect all points in N to G in such a manner that the total length of the required links is minimized.

If G degenerates to a single point, our problem is reduced to the well-known Steiner tree problem. Therefore, we have a generalized version of that problem, and may refer to the optimal solution as the Generalized Steiner Minimal Tree problem (GSMT). Note that the links incorporated in the solution may or may not form a spanning tree by themselves, but together with G, they do span $N \cup V$, and if G is a tree, then the GSMT is also a tree. In our discussion, we may refer to G as a single supernode (to which we assign the index 0), so that in a sense the GSMT is a tree, even if G contains cycles. However, there is no reason to expect that the GSMT will be a proper Steiner tree (let alone an SMT) for $N \cup V$.

Simple and Compound GSMTs: If N is ultimately connected to G through one link exactly, we call the GSMT a Simple GSMT. All other cases are named Compound GSMTs and are actually combinations of partial, simple GSMTs. For example, in Figure 1 a case of a compound GSMT is depicted but it can be broken down to four simple components, namely the connections of $\{1,2,3,4\}$, $\{5,6,7\}$, $\{8\}$ and $\{9,10\}$ to G.

We refer to the case where $|N| = 1$ as "the basic case." In order to solve it, we have to find the nearest arc of A to the node (where each arc includes its endpoints); if the connection is not through an endpoint, it must be by a perpendicular link. Thus, in order to solve the basic case, we have to check up to $|A|$ arcs.

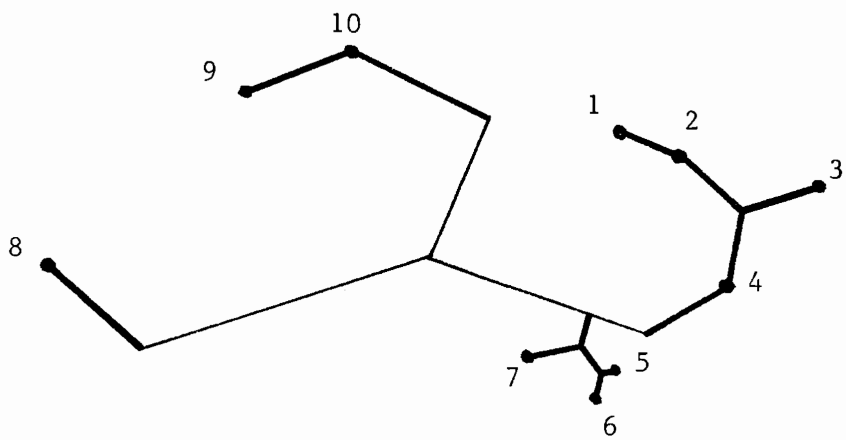


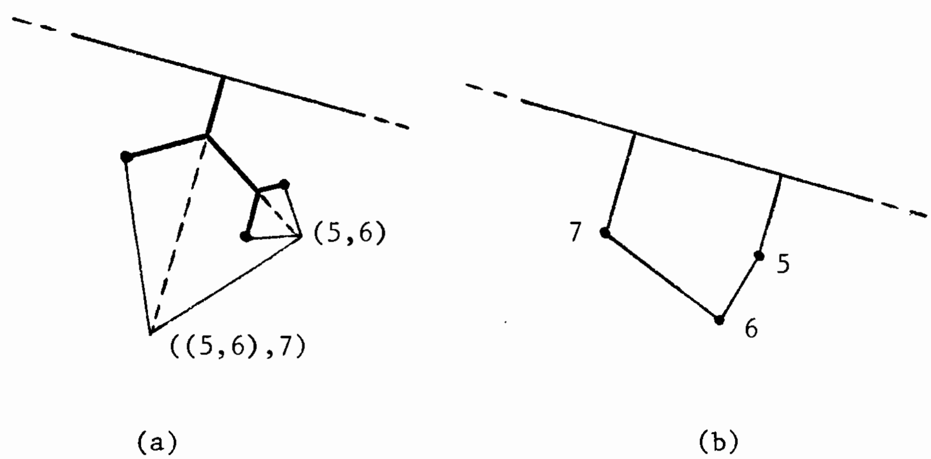
Figure 1

For some less trivial cases, we need the extended Steiner construction, which is applicable to a set of $M \subseteq N$ of $m \leq n$ nodes connected to G by a simple GSMT with m - 1 Steiner points exactly. This is actually a full Steiner tree, or a GFST. Note that for m = 1 a GFST is a single link.

According to Cockayne [1], a full Steiner tree can be represented by a notation which indicates a pairing order of nodes of N (where a pairing implies representing two points by another point on the apex of an equilateral triangle based on the segment associated with the pair). Cockayne also showed that node n (or any other node) can always be left alone and thus become an endpoint of the segment which represents the whole FST at the end of the pairing process. (As an example, take a Cockayne notation such as ((1,2), (3,4)), 5; this would indiate representing 1 and 2 by (1,2), 3 and 4 by (3,4), and finally (1,2) and (3,4) by ((1,2), (3,4)), which, with 5, forms a segment. What we do in our case is simply to leave G, the $(m + 1)^{th}$ node, as such an endpoint, and connect it to the other node--representing M--as per the basic case (thus locating the exact point to which we should make the connection). Note that it is very easy to apply this extended Steiner construction not only to cases where G is a single segment, but also to any network, as long as we assume that we are looking for a GFST with a given configuration.

As in the regular Steiner construction, degeneracy may occur, thus indicating that a certain configuration does not exist for $M \cup G$.

<u>An Example</u>: Let $N = \{1,2,\ldots,10\}$, as depicted in Figure 1, and let $M = \{5,6,7\} \subset N$. It can be shown that for this subset, the GFST depicted in



<u>Figure 2</u>

Figure 2 is the GSMT, and the Cockayne notation associated with it is ((5,6),7), 0. Figure 2a illustrates the extended construction, where nodes 5 and 6 are represented by (5,6), which is represented in turn, with 7, by ((5,6),7); ((5,6),7) is connected to G perpendicualarly, as per the basic case. (Part b of the figure describes the generalized Steiner polygon for this subset, for the discussion of which the reader is refered to [8].)

At this stage, we can present an algorithm which solves our problem in finite time, as follows: Look for all possible subsets $M \subseteq N$ which can be connected by GFSTs to G; each such M, along with the minimal GFST associated with it implies a super-network (which includes G and the minimal GFST), and a set $N - M$ of nodes which we still have to connect (unless $N - M = \emptyset$); clearly, if we continue in a similar manner (find a subset, etc.), we must ultimately obtain a solution, and the best of these solutions is the required optimum.

It can be shown that the proposed algorithm is exponential in $|N|$, and it is by no means presented here as an efficient algorithm. It is comparable, however, to the regular Steiner tree algorithm [1], since both "check" all the partitions of N.

Now, our premise is that G consists of a finite number of connected segments, and for any subset $M \subseteq N$ and full topology (or configuration), we can easily locate the best segment to connect through. However, in the choice of M, we can sometimes save much time by intelligent inspection, and by the use of the Steiner polygon [1], which we reintroduce here for completeness.

<u>The Steiner Polygon (Cockayne)</u>: For a set of N points, connect all $n(n - 1)/2$ pairs by straight segments, and let P_o be the convex hull polygon of all the segments (and N). Obviously, P_o is formed by a subset of the segments, and a subset of N is on its boundary. This completes our initial preparations, and we proceed with stage 1.

In stage i (for $i = 1, \ldots$), P_{i-1} is given. If for any edge on the boundary of P_{i-1}, say $\overline{k,\ell}$, there exists a point $m \in N$ such that $\Delta m,k,\ell$ does not contain any other point of N, and such that $\angle k,m,\ell > 120°$, then P_i is obtained from P_{i-1} by dropping $\overline{k,\ell}$ and incorporating $\overline{k,m}$ and $\overline{m,\ell}$ in its stead. If no such boundary edge exists, P_{i-1} is the Steiner polygon.

Now, let P_o denote the convex hull polygon of N (or of $M \subseteq N$, as the case may be); let P be its Steiner polygon; and finally, let Q denote the Steiner polygon as defined for $N \cup V$ (or $M \cup V$). Then:

Theorem 1: When solving for any set M, only those edges of G which are accessible from P_o by straight uninterrupted lines need be considered.

Proof: Trivial. []

Theorem 2: If Q is partitioned by various chains of edges in G to some disjoint faces, the GSMT can be obtained by solving separately for the edges of each such face and the subset of N which is contained in it.

Proof: See Theorem 6 in [8].

Theorem 3: If P_o is intersected by G to some disjoint parts, it suffices to solve the problem separately for the points of N in each of these parts.

Proof: Clearly the condition is sufficient (but not necessary) for Theorem 2 to hold. []

Note: Theorem 1 will then serve to identify the edges of A which need be considered for each subset of N.

Theorems 1 and 2 may be used in a straightforward manner to facilitate the solution procedure. Theorem 3 is not strictly necessary since its applicability implies the applicability of Theorem 2; however, when solving manually, it is sometimes clear at a glance that Theorem 3 applies and it is easier to apply than Theorem 2.

2. The Network Connection Problem

In this section we describe our current problem, and prove that the best "basic solution" (defined below) is our optimum. This makes possible the extension of the technique described above for the solution. We proceed with some formal definitions and results.

The Problem

Let $G = \{G_i(V_i,A_i)\}_{i\in I=\{1,2,\ldots,n\}}$ be a set of internally connected but mutually disjoint networks on a Euclidean plane, where V_i is the set of vertices of G_i, spanned by a set of straight arcs A_i (connecting pairs of vertices of V_i), and finally let $V = \{V_i\}_{i\in I}$ and $A = \{A_i\}_{i\in I}$. We assume that any intersection of two arcs implies a vertex ("planar graph").

Definition 1: A set of (new) links $L = \{L_j\}_{j\in J}$ which connect all the networks together is called a solution.

If we assign a cost to any possible link which may be included in L, our problem is to choose links for L so as to minimize its total cost. In this paper the cost of a link L_j is taken as its Euclidean length--$d(L_j)$, and we are looking for the set which the "total length," namely, $\Sigma d(L_j)$, is minimized. We refer to this case as the Steiner case.

A solution may be a tree, a forest, a network or a set of disjoint networks. In the Steiner case, however, only trees and "simple forests" (defined below) need be considered.

Definition 2: A set of disjoint trees (i.e., a forest) is called a simple forest if there is no cycle through the networks (though any network G_i may have internal cycles).

If all the networks in G are trees, and the solution is a tree or simple forest, then $G \cup L$ is also a tree. If we regard networks G_i as supernodes, then our problem is to find a Steiner minimal tree for those supernodes. Obviously this need not necessarily be a regular Steiner tree.

Definition 3: A chain of links in L which connects two networks $G_i, G_j \in G$, without passing through any other network of G, is called a <u>direct</u> chain.

Note that a direct chain need not necessarily be a single link, since it may pass through Steiner points.

Definition 4: A point $x \in G - V$ (or $A - V$) is called an inner point. In contrast, a vertex of V may be referred to as an endpoint.

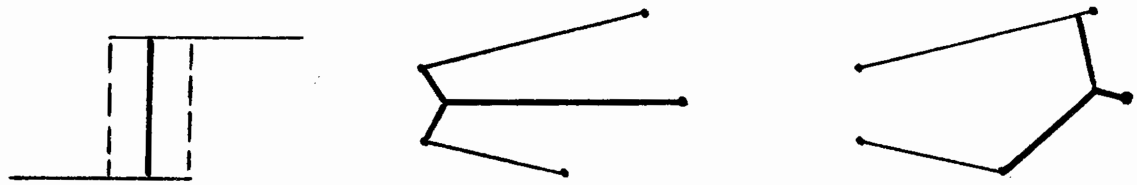
Definition 5: A direct chain is called a basic (direct) chain if it starts or ends at endpoints. In contrast, a direct chain connecting two inner points is called a nonbasic chain.

Definition 6: A solution where all the direct chains are basic, is called a basic solution. If one chain or more are nonbasic, the solution is nonbasic.

Definition 7: A solution where all angles between adjacent links of L are 120° at least, and all angles subtending links of L and adjacent arcs of A are not acute is called a stable solution. Else, it is called unstable.

Note that in a stable solution links of L adjacent to the arcs of A at inner points must form two right angles there, and at Steiner points we have three angles of 120° (as in regular Steiner trees).

If all angles between links and adjacent arcs are strictly obtuse, we may refer to the solution as strictly stable. This cannot happen if any connections are made through inner points. Hence, a strictly stable solution must be basic. We illustrate these concepts in Figure 3, where part a depicts a nonbasic but stable solution (with basic ones in dashed lines), part b depicts an unstable but basic solution, and part c depicts a better solution than that of part b, which is strictly stable and basic.



(a) (b) (c)

Figure 3

Lemma 1: The optimal solution is stable.

Proof: Optimal implies locally optimal which in turns implies stable. ▯

The set of all the possible solutions is uncountable--even if we confine ourselves to trees or simple forests--since there are uncountably many inner points through which connections can be made. If we confine ourselves further to stable solutions (as we may, using Lemma 1), the number of solutions is still not necessarily countable--as is illustrated by Figure 3a. However, the number of basic and stable solutions is finite.

One of the main results of this paper is that we can confine ourselves to this finite set without risk of losing the optimum. In order to show that, however, we need some more preparation, such as the following inductive definition of parallel shifts (see Figure 4):

Definition 8: For a (basic or nonbasic) direct chain of one link, a parallel shift is defined by moving the link to a parallel position at a distance of ε ($\varepsilon > 0$) from the original position, in such a manner that the shifted link will still connect the same arcs of A as before (through inner points or endpoints). (Note that such a shift cannot alter the connectivity of our system. Also note that the parallel shift can be executed in two directions, and there is always an ε small enough to make it possible. These observations

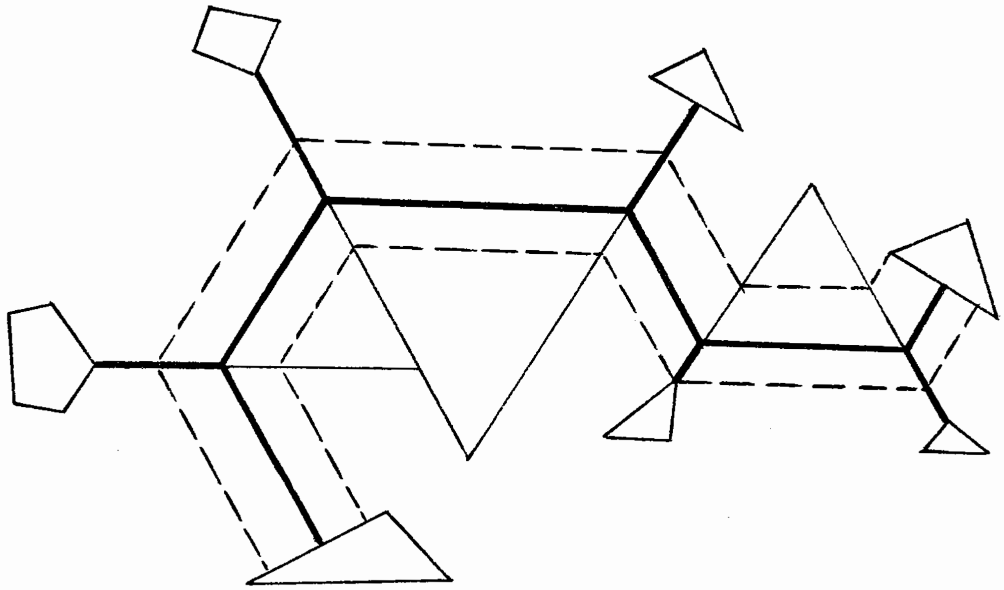


Figure 4

will also hold for the general parallel shift--for chains of more than one link--as defined below.) Now assume that the parallel shift has been defined for any (basic or nonbasic) direct chain of k - 1 links (k = 2,3,...). It is required to define it for chains of k links. To that end, choose any Steiner point along the chain (at least one exists for k ⩾ 2), and there are three links incident to it: two links belong to the direct chain, and a third link to a point of G ∪ L (say, c); extend the third link as far as possible within conv(G), in the direction away from c, without crossing G ∪ L again or crossing any other extension of the same kind associated with the two parts of the chain (as divided by the selected Steiner point). Now, each part of the chain is a chain of k - 1 links or less, so we have defined (and can execute) parallel shifts for them. Furthermore, if we choose the same ε and direction for each part, the two shifted chains will still form one connected chain, and they will be connected at a point along the extended link associated with c. However, ε must be chosen in such a manner that on neither subchain will a link be connected out of the segment or ray it was connected to before (be it an arc of A or an extended link associated with one of the Steiner points). Finally, the new breaking points (angles) of the shifted chain replace the old Steiner points, and they are all connected to $G \cup L$ through their respective extended links to the same connection points as before (such as c).

A parallel shift can always be executed, and in two directions. However, in each direction there is a maximal value ε can assume.

Definition 9: A parallel shift with the maximal value of ε in its direction is called a maximal parallel shift.

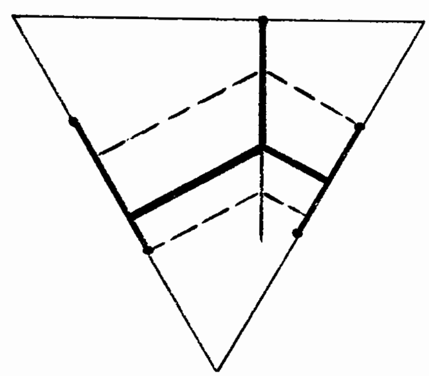
A maximal parallel shift performed on a nonbasic chain connecting, say, G_i and G_j, must result in one of the following outcomes: (a) the shifted chain becomes basic, i.e., at least one end of it merges to a vertex of V_i or V_j, or (b) the shifted chain is split into two subchains, i.e., one of the Steiner points along the chain is merged into a point of G_k for some $k \neq i,j$-- in this case one of the following must occur: (i) the subchains are both basic; or (ii) the subchains are connected to an arc of G_k at acute angles, and the solution becomes unstable; finally (iii) it may occur that two of the Steiner points (one or both of them on the chain) merge with each other--and again the solution becomes unstable, since angles of 60° are formed at the rank four merged point!

We are now ready to state and prove Theorem 4, our main result for this section:

Theorem 4: If an optimal solution is nonbasic, then there exists a basic solution which is optimal too.

Corollary: The best basic solution is optimal.

<u>Proof</u>: If we can show that parallel shifts performed on any stable and nonbasic chain do not alter the value of the objective function, then, by performing maximal parallel shifts on the nonbasic stable chains of our solution, we either reduce the number of nonbasic chains by one for each maximal parallel shift, or achieve an unstable solution. In the former case, we will (utlimately) get a basic solution, while in the latter case--by Lemma 1--our nonbasic solution was not optimal to begin with! We now proceed to show (inductively) that this is the case: for nonbasic stable chains of one link, our result is obvious, since the distance between parallel lines is constant. For any chain with two links, the angle between the two arcs of A connected by it must be 60° (see Figure 5). Draw a perpendicular to the



<u>Figure 5</u>

third link connected to the Steiner point through its connection to G ∪ L, and extend the two arcs as may be required, and we obtain an equilateral triangle, with the Steiner point in it, and the three links from the Steiner points are perpendicular to the sides of the triangle. Executing a parallel shift means choosing another location for the Steiner point but still within the triangle, and it would not change the sum of the three distances of the Steiner point from the sides of the triangle--but this sum is the contribution of the chain and its Steiner point to the objective function, and we see that it remains unaltered! Having proved for stable and basic chains of up to two links, we proceed by induction, dividing any chain of k links to two smaller chains (by a properly chosen perpendicular "cut"), to complete the proof. ▯

Using the results of the network augmenting problem we can now solve the network connection problem in finite time. this is true because the optimal solution must be composed of partial full Steiner trees, each spanning some vertices of V--belonging to disjoint networks--plus at most one inner point of another network. Naturally we would have to check out all the legitimate partitions and all the possible choices of the vertices--a "highly" exponential task, but a finite one.

In the next section we discuss some shortcuts and improvements to this procedure.

3. Some Shortcuts and Improvements

In our procedure described above, similar to the algorithm in [1], we have to generate all the possible partitions of G--an awesome task. Furthermore, after we have a partition, then for each part we still have to choose the connection points p_i to be connected by a GSMT (which should also be full, or another partition will yield the minimum more conveniently). This leads to a prohibitive number of possibilities.

What we can do is (i) use the Steiner polygon and Theorems 1, 2 and 3, properly extended, to rule out some partitions in advance, and (ii) to find an efficient procedure to the connection points choice problem.

As for the Steiner polygon, if we define it for V, we may be lucky enough to get a partition of the problem, if any network G_i connects opposite sides of the polygon. If may also impose a cyclic order when used for G or for any subset of it.

To extend Theorem 1, note that we can now apply it for each of the networks separately. To do that for, say, G_i, all we have to change is that P_o should be the convex hull of $\{V_j\}_{j \neq i, j \in M}$. To extend Theorem 2, again for each network separately, define P as the Steiner polygon for the same subset and Q as the Steiner polygon for V.

We devote the remainder of this section to the problem of choosing the connection points p_i, $\forall i \in M$. W.l.o.g. we assume that we are only interested in a simple, minimal, and full Steiner tree connecting the networks of the subset M. If it is not minimal, it is not part of the optimal solution; if it is not full, another partition will take care of it, through the parts. Our first case will be a convex one, and we show how it can also serve in a more general context. To continue, we define some old/new concepts:

<u>Definition 10</u>: Any subset of L (whether L is optimal or not), which is a Steiner tree in itself, is called a simple generalized Steiner tree, or a simple GST.

<u>Definition 11</u>: A simple GST connecting m networks with m - 2 Steiner points is called a generalized full Steiner tree, or GFST (note that a full Steiner tree must be simple to begin with).

<u>Definition 12</u>: If all the sides of conv(G_i) are in A_i, then G_i is called a <u>convex network</u>.

When defining a convex network, we are motivated by an implicit assumption that we connect the networks by links located outside their respective convex hulls. This assumption can be dropped easily enough, if necessary. E.g., the example given in section 1 does not have this property.

<u>Definition 13</u>: The locus of the points which may be connected in a stable

manner to a vertex $v \in V$ is called the cone of v, or cone(v). |cone(v)| measures this set, generally in degrees.

Note that cone(v) may be empty sometimes (and thus we can save some effort by avoiding any tentative connection to it). If $G_i = V_i = v$, i.e., a degenerate network, |cone(v)| = 360°, and we can connect to it from any point on the plane as long as we do not cross some other network on our way. In most cases, however, |cone(v)| ⩽ 180°, and equality occurs if and only if rank(v) in G_i is one.

Now, let P_o be defined as in the extension to Theorem 1, as the convex hull of $\{V_j\}_{j \in M, j \neq i}$, then clearly nodes $v \in V_i$ such that cone(v) does not intersect with P_o are not candidates to be the connection to G_i--this may reduce our task considerably. Furthermore, if this happens for two adjacent nodes, say v and w which are connected by an arc $\overline{v,w} \in A_i$, then two possibilities exist: (i) if the two cones are to the same side of P_o, then no stable connection is feasible to the arc $\overline{v,w}$ either; or (ii) if P_o is "sandwiched" between cone(v) and cone(w), only connections to $\overline{v,w}$ need be contemplated. Incidentally, if it happens for more than one network in M, that it can only be connected to the others through inner points of arcs, then by virtue of Theorem 4, we need not consider any partition with M as part of it.

GFSTs For Convex Networks

For a set M of m ⩽ n convex networks, which w.l.o.g. we index from one to m, and a given full topology, the following lemma implies that solving for the Steiner tree associated with the full topology is a convex problem.

Lemma 2: For a set M of convex networks and a full Steiner tree configuration, then one of the following must happen: (i) a stable GFST exists and is optimal relative to M and the given configuration, or (ii) a GFST for M and the configuration is either unstable or nonexistent.

Proof: Since the configuration is given, what we must do is locate the m connection points, each on a convex set, and the m - 2 Steiner points--also a convex problem. Hence ours is a convex problem, and as such it may have a stable basic GFST solution. Or, it may turn out that a Steiner point merges with a connection point resulting in a simple GST, but not a full one. Or, two Steiner points may merge, thus violating the stability requirement (and indicating that another full topology is superior (see [9])). □

The following lemma is our motivation for discussing the special convex case in detail.

Lemma 3: For a set M of not necessarily convex networks and a full Steiner configuration, then if a GFST exists for the convex hulls of the networks connection problem, its value is a lower bound on the value associated with the same configuration and the original networks.

Proof: Clearly if the "convex" solution is feasible for the real problem, it is also optimal locally (i.e., for the given configuration); else, we may have to connect through "further away" connection points, with a higher total length. □

Thus armed with a stopping criterion (a stable GFST at hand), we proceed to identify the stable GFST, if one exists, for the convex case.

(a) Choose (arbitrarily or by an appropriate heuristic) a set of tentative connection points $\{p_i | p_i \in Vi\}_{i=1,..,m}$.

(b) For $\{p_i\}$ and the full topology construct a representative segment [1,9] with p_1 as an endpoint and, say, x_1 as the other endpoint (representing $\{p_i\}_{i=2,\ldots,m}$).

(c) In $\text{conv}(G_1)$ look for any point $\in V_1$ nearer to x_1 than p_1. If there is one, it becomes the new tentative connection point to G_1, p_1.

(d) Repeat steps (b), (c) for $p_2, p_3, \ldots, p_n$, etc., until for a full cycle no point p_i is changed (finite convergence can be proven here, due to the convexity of the problem and its structure).

(e) If all the segments $\overline{p_i, x_i}$ are connected in a stable manner, go to (f). Else, shift the (one!) unstable point to an appropriate inner point so that $\overline{p_i, x_i}$ will be perpendicular to the arc it is now connected to. Check if the other m - 1 connections remain stable. (If not, a direction of descent is implied, and since our problem is convex we shall ultimately converge to a "stable" choice of connecting points. Below we discuss this case in some further detail.) Go to (f) with the set $\{p_i\}_{i \in M}$ which makes the segment shortest.

(f) If the segment implies a GFST, stop; else by Lemma 2 the M/configuration pair can be discarded without loss of optimality.

(The reader is referred to [9] for a more elaborate discussion of the relationship between segments such as $\overline{p_i, x_i}$ and the resulting Steiner tree. Note that some such segments do not yield Steiner trees.)

This completes our discussion of the convex case. As for the nonconvex case, we have Lemma 3, but we may still want the optimal solution if it does not coincide with the one for $\{\text{conv}(G_i)\}_{i \in M}$, i.e., some of the $\{\text{conv}(G_i)\}$ connection points do not belong to $\{G_i\}$. (This can only happen at inner connections.) In this case we have to check for all the candidate connection points implied for each "unconnected" network. Note, however, that we still require at most one connection to an inner point, so we will not have a problem with more than one network, say G_j, in each iteration.

Since the case of an inner connection may arise for both the convex and nonconvex cases, it merits some more elaboration. By the nature of the instability we may know to which arc we wish to connect perpendicularly (for instance, in the convex case we always know that). But, the angle of the perpendicular to this arc, together with the full topology dictates the angles of all the other m - 1 connections! Hence we can easily identify all the candidate connection points (which are also unique in the convex case). If this is impossible, and it is also impossible not to connect to an inner point of that specific arc, the pair M/configuration is not viable, and may be summarily discarded.

To return to the nonconvex case, locating the conenction points thusly may not result in unique points, which calls for a slightly more extensive search. However, if only one network is nonconvex, and if we find a stable connection to it through a connection point p_j such that for the point x_j associated with it p_j is the nearest connection in the nonconvex network G_j, we are home free!

Incidentally, if an inner point's connection is indicated, we do not strictly require the Steiner construction to construct the GFST, by may resort to a solution by a set of linear equations with slopes as per the required angles. This may be a little easier to do; however, the Steiner construction can also be performed by solving linear equations recursively, so the difference here is quantitative, and not qualitative.

4. Examples and Discussion

Our first example is a three networks case depicted in Figure 6. By observing the respective cones of the vertices we see that vertices 2, 3, 5, 9, 10 are not candidates to be connection points. Likewise, the arcs $\overline{2,3}$, $\overline{4,5}$, $\overline{5,6}$, $\overline{8,9}$ and $\overline{9,10}$ can be excluded.

We first try to connect by a GFST. Although it is obvious that 7, 4 and 1 should be the connection points in this case, we choose 6, 8 and 1. Obviously, this cannot lead to stability and indeed 7 is nearer to (1,6), and

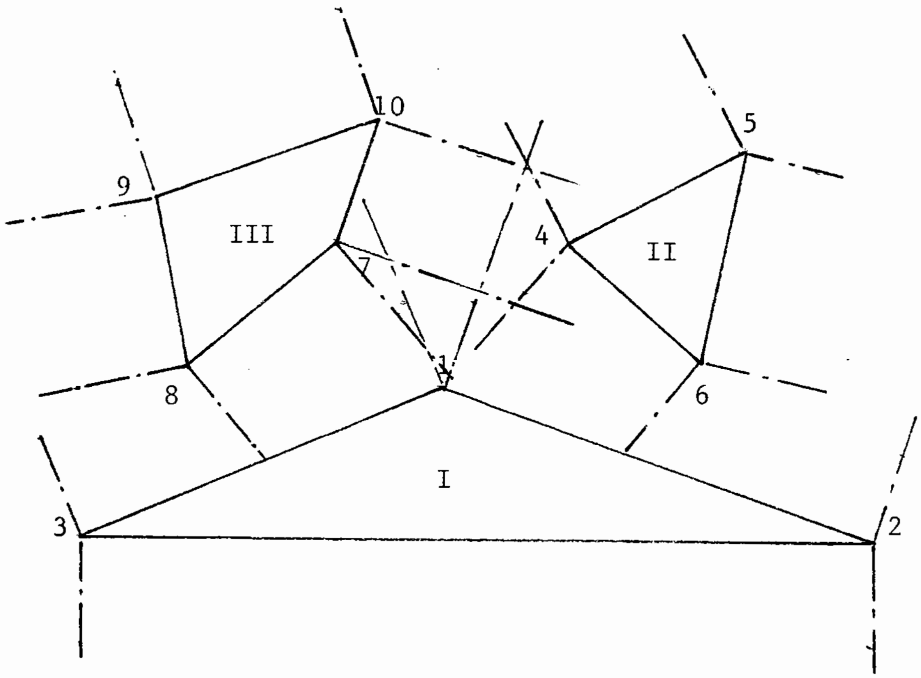


Figure 6

adopting it we then find that 4 is nearer to (1,7), leading to a stable representation associated (in this case) with a GFST, as depicted in Figure 7.

However, if we connect 8 to $\overline{1,3}$ and 6 to $\overline{1,2}$, as implied by the partition $\{I,II\}$, $\{I,III\}$, we get a slightly better result (see dashed lines in the figure) which is the optimal solution in this case. (Note that the connections of 4 and 7 to $\overline{1,2}$ and $\overline{1,3}$, respectively are (a) not stable, and (b) worse than the GFST. This means that a greedy improvement scheme would fail here, as the problem implied by it is <u>not</u> convex. This is the counterexample to the extension of [9]'s greedy partitioning algorithm even to our "convex" case.)

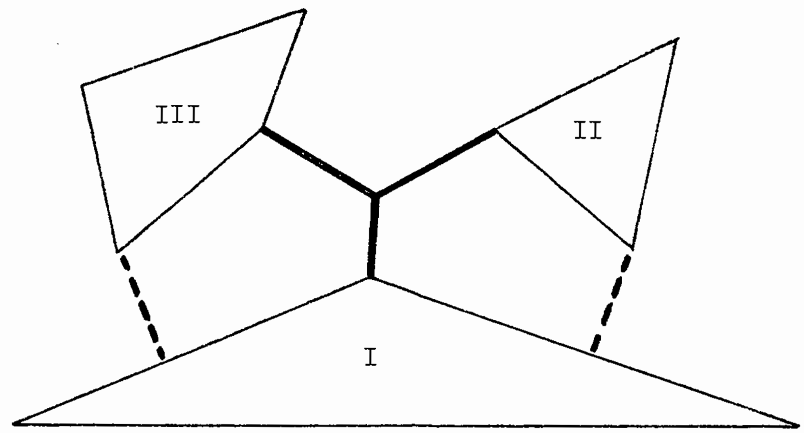


Figure 7

Our second example illustrates the use of Theorems 1 and 2, depicted in Figure 8, where we have, again, three networks.

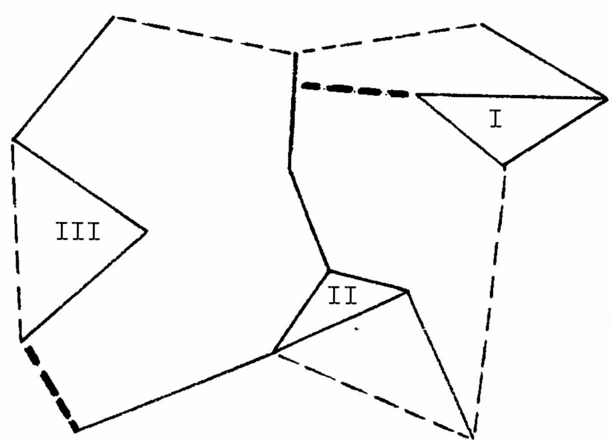


Figure 8

The Steiner polygon for V is given in a dashed line, and we see that network II cuts it to two parts, so by Theorem 2 we may solve separately for the connection of I and II and of II and III. Clearly the optimal connection of conv(I) and II is the perpendicular shown by a thick dashed line and it is feasible--so we adopt it. A similar observation holds for the other connection.

Finally, Figure 9 depicts a case where the stable segment implies a degenerate GST, and hence a further partition.

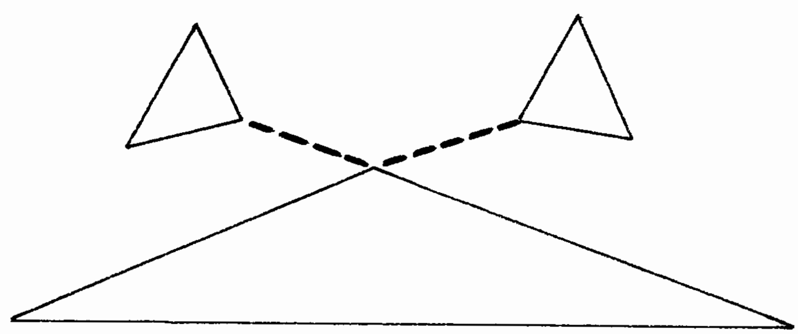


Figure 9

Conclusion

We have demonstrated a finite procedure to solve the Euclidean network connecting problem, by extending to it known techniques used in the Steiner tree problem, including the Steiner construction which can be executed by a ruler and compass. The problem may be extended to connecting networks with rectilinear distances, or networks embedded within graphs--cases for which versions of the Steiner tree problem exists. All these problems may be generalized by assigning costs to the edges according to their flows (see

Gilbert [3], or Trietsch and Handler [7]). In a subsequent paper we intend to show to what extent our results can be generalized for this case and to discuss some heuristics associated with it. Note that we do not "need" Steiner points to achieve NP-completeness if we assign flow dependent costs to the arcs (see a virtually equivalent result in [5] where a budget is used instead of the variable costs). In this case, even if we do not allow Steiner points, and thus revert to a regular network design problem, it is not just NP-complete but one of the toughest problems in that equivalence set (even though they are all "equal"). This calls for the development of heuristics and shortcuts.

Finally note that the Gilbert and Pollak conjecture [4], if true, certainly holds for the Euclidean network connection case; similarly its generalized version [10] would hold for the flow dependent costs case.

References

[1] E. J. Cockayne, On the Efficiency of the Algorithm for Steiner Minimal Trees, SIAM J. Appl. Math. 18 (1970), pp. 150-159.

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

[3] E. N. Gilbert, Minimum Cost Communication Networks, Bell System Technical Journal 46 (1967), pp. 2209-2227.

[4] E. N. Gilbert and H. O. Pollak, Steiner Minimal Trees, SIAM J. Appl. Math. 16 (1968), pp. 1-29.

[5] D. S. Johnson, J. K. Lenstra and A. H. G. Rinooy Kan, The Complexity of the Network Design Problem, Networks 8 (1978), pp. 279-285.

[6] Z. A. Melzak, On the Problem of Steiner, Canad. Math. Bull. 4 (1961), pp. 143-148.

[7] R. C. Prim, Shortest Connection Networks and Some Generalizations, Bell System Technical Journal 36 (1957), pp. 1389-1401.

[8] D. Trietsch, Augmenting Euclidean Networks--The Steiner Case, to appear in SIAM J. Appl. Math.

[9] D. Trietsch, An Improved Algorithm for Steiner Trees, discussion paper No. 635, Center for Mathematical Studies, Northwestern University.

[10] D. Trietsch and G. Y. Handler, The Gilbert and Pollak Conjecture--A Generalization, to appear in Networks.