Journal of Computer and System Sciences **63**, 432–448 (2001)
doi:10.1006/jcss.2001.1785, available online at http://www.idealibrary.com on IDEAL®

Distance Realization Problems with Applications to Internet Tomography

Fan Chung

University of California, San Diego, La Jolla, California; and Telcordia Technologies, Morristown, New Jersey 07960

Mark Garrett[1]

Telcordia Technologies, Morristown, New Jersey 07960

Ronald Graham

University of California, San Diego, La Jolla, California 92093

and

David Shallcross[1]

Telcordia Technologies, Morristown, New Jersey 07960

Received January 10, 2000; revised December 12, 2000

In recent years, a variety of graph optimization problems have arisen in which the graphs involved are much too large for the usual algorithms to be effective. In these cases, even though we are not able to examine the entire graph (which may be changing dynamically), we would still like to deduce various properties of it, such as the size of a connected component, the set of neighbors of a subset of vertices, etc. In this paper, we study a class of problems, called *distance realization problems*, which arise in the study of Internet data traffic models. Suppose we are given a set S of terminal nodes, taken from some (unknown) weighted graph. A basic problem is to reconstruct a weighted graph G including S, with possibly additional vertices, that realizes the given distance matrix for S. We will first show that this problem is not only difficult bu the solution is often unstable in the sense that even if all distances between nodes in S decrease, the solution can increase by a factor proportional to the size of S in the worst case. We then proceed to consider a weaker version of the realization problem that only requires the distances in G to upper bound the given distances. We will show that this weak realization problem is NP-complete and that its optimum solutions can be approximated to within a factor of 2. We also consider several variants of

[1] Research supported in part by DARPA Contract F30602-97-C-0188.

0022-0000/01 $35.00

these problems and a number of heuristics are presented. These problems are of interest for monitoring large-scale networks and for supplementing network management techniques.

1. INTRODUCTION

In the rapidly growing world on Internet infrastructures, we face many challenging new mathematical problems. These arise in part because the usual assumptions made in problems of this general type may no longer hold. For example, many typical questions dealing with massive data sets often involve networks or graphs of prohibitively large sizes so that the (exact) number of nodes is no longer a useful parameter. Instead, only partial information can be obtained, for instance, by setting up monitors at a relatively small subset of the nodes. From the monitors, data can be collected and examined. The problem of discovering the detailed inner structure of the network from a collection of "end-to-end" measurements can be seen as a type of *inverse problem*, analogous to those arising in tomography, but with a strong discrete flavor. For example, problems of interest include checking connectivity, finding largest components, tracking data traffic, mapping usage patterns, assessing the performance of software and hardware, and dealing with a variety of security and reliability issues.

In this paper, we focus on several fundamental problems in combinatorial algorithms, which we call *distance realization problems*. In the next section, we will discuss the basic setup of distance realization problems in Internet tomography and the advantages and disadvantages of this approach. In Section 3, we will formulate several problems of reconstructing the graph G based on given distances among vertices in a (small) subset S of the nodes. We will briefly survey the history of the distance realization problems and the related heuristic algorithms. As we will see later, there are substantial obstacles for deriving solutions to achieve the exact distances given in the distance matrix of S. We will show (in Section 4) that there are examples of distance matrices for which the solutions vary in a dramatic way (by a factor of the size of S) when the distance matrix only changes very little. Because of the difficulty of deriving approximations, we consider a variant, called the weak distance realization problem. This is to find a graph G containing S with minimum total edge length so that the distances in G between nodes of S are greater than or equal to the given distances. We will show that the weak distance realization problem is NP-complete but that there are approximation algorithms which achieve solutions to within a factor of 2 of the optimum (see Section 5). Another variant that we call *rooted weak realization* is a weak realization with an additional requirement that the exact distances from nodes of S to a specified root are archieved. It will be shown in Section 6 that the rooted weak realization problem is also NP-complete. We present several heuristic algorithms and examine some related problems such as the Euclidean Steiner problem, the graph Steiner problem, universal graph problems, and the realization problems with given incidences of paths and edges. The implementations of heuristics for distance realization problems in Internet tomography will be dsicussed in the last section.

In addition to the applications to Internet tomography, the distance realization problems that we discuss here turn out to have a large number of applications in computational biology (e.g., constructing phylogenetic trees from genetic distances among living species), analyzing clustering, and classification [3, 4, 6], for example.

2. DISTANCE REALIZATION IN INTERNET TOMOGRAPHY

We consider a prototype monitoring system in which a set of monitors are placed throughout the Internet or within a regional network. The measurement technique is *sparse active monitoring* where monitors create their own traffic, but at a very low level. The monitors measure delay and loss properties of the network by transmitting packets to each other. The problem of interest is to infer network topology from end-to-end nonintrusive measurements (Fig. 1). This approach is different from the direct Internet route-tracing methods and is independent of traditional network management infrastructures. This independence provides several advantages that can contribute to more robust network management generally and allow network monitoring when normal network management is absent or compromised. (The reader who whishes to move directly to the technical part of distant realization problems may skip this section.)

There are two conventional approaches to keep track of network topology. The oldest, and still pervasive, method is to manually construct a database as the network is built. Network topology data in commercial telephone and Internet service providers is notoriously inaccurate due to human error in tracking changes as the network evolves. A more sophisticated method that is now emerging in standards and practice is network autodiscovery, where a network management system finds and tracks the identity and location of network elements through a protocol. Autodiscovery places some level of processing load on the network elements and creates excess traffic in the network. It also requires secure access permissions between the network elements and the management system. An exception, of course, is the ICMP (Internet control message protocol), which, if enabled, allows any Internet host to query routers for their identity and a timestamp.

Autodiscovery is a reasonable basis for automated network management, but can still be inoperative or inadequate in some situations. First, packet networks today are composed of many technologies, usually partitioned into several layers. For example, a set of IP (Internet protocol) routers may be considered to be at network layer 3, while there can be packet congestion, failure, and errors in layer 2 ATM network (asynchronous transfer mode) or layer 1 of optical fiber networks that are not visible to layer 3 network management tools. For example, ICMP-based ping and traceroute are commonly used to identify IP routers on a path, but these programs will not see ATM, frame relay, MPLS (multiprotocol label switches), or optical-layer switches.

A second reason that autodiscovery may be insufficient is that network management mechanisms must be consistently built into all of the network elements and that requires standardization, acceptance by multiple vendors, and time for

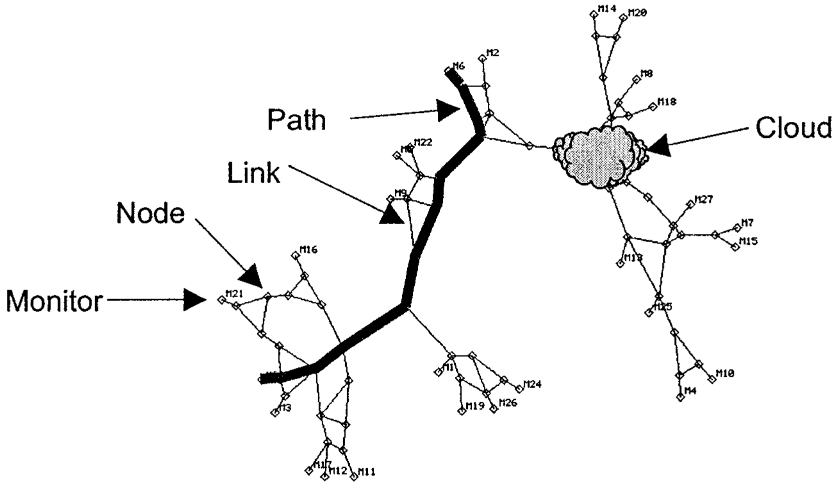

FIG. 1. An example of a network topology.

implementation. Networks can reasonably be expected to contain technologies that are either too new or too old to be compatible with any given autodiscovery (or general network management) system.

Third, the goal of network monitoring may be to evaluate a network that is not controlled by the observer. In a commercial context, the quality of an operator's service may depend on the transport provided by another operator, whose network is part of the end-to-end transport path. In a military context, the observer may wish to analyze a foreign network inconspicuously. Finally, networks may be subject to intrusion or attack (or simply software bugs) that compromise some network elements or connectivity.

Our approach based on distance realization methods presents an alternative or supplement to the conventional monitoring methods. Our solutions can be useful not only for inferring network topology, but also in tracking routing changes or locating network congestion and faults, detecting network intrusion patterns and possibly other security applications.

Nevertheless, our network monitoring system has certain fundamental limitations and properties. If the monitors are well distributed, most of the backbone links will be traversed by monitor packets and can then be discovered. As we move from the backbone to the edges of the network, it becomes more likely that monitor packets will not traverse some network links. In general, areas of a network that do not lie on a path between two monitors cannot be discovered by this system. A part from the obvious implication that the discovered network is incomplete, it is also true that a different choice of monitor locations will change the portion of the network that is discovered.

Another limitation arises when two links are traversed by identical sets of paths, such as two edges incident to a node of degree 2. Suppose we delete elements not traversed by monitor packets, and combining serial edges, we construct a reduced graph representation of the network. Although it is incomplete, the reduced graph is closely related to the actual network, and the discovery of the reduced graph may be enough information to be useful for network monitoring.

In the prototype monitor system deployed at Telcordia Technologies, four different inference schemes are implemented for topology discovery including the distance realization methods. Multiple independent algorithms are used for validation and correction at various intermediate stages or on a subset of paths and regions. Further discussion on implementation of distance realization algorithms is included in Section 7.

3. SEVERAL DISTANCE REALIZATION PROBLEMS

We consider a weighted graph Γ where each edge is associated with a positive value which can be regarded as its *length*. For a path p in Γ, the length of P is just the sum of the lengths of the edges in P. The distance between two vertices u and v in Γ, denoted by $d_\Gamma(u, v)$, is defined to be the length of a shortest path joining u and v in Γ.

In a graph Γ, let S denote a subset of nodes, called terminal nodes. Suppose that the distances of all pairs of terminal nodes are known but the edges of Γ as well as their lengths and topology are unknown. The general problem of interest is to reconstruct Γ, based on the information about pairwise distances of the terminal nodes. Of course, such a general problem might not be well defined since there can be more than one graph satisfying the distance constraints. For example, we can construct a graph G with node set S and with $\binom{|S|}{2}$ edges so that the length of the edge joining u and v is just $d_\Gamma(u, v)$. Such a graph certainly satisfies the distance constraints, but it is unlikely to be the network topology that we seek to discover. Suppose Γ contains some vertex x which is not a terminal node and x has degree 2 (i.e., there are only two neighbors y and z of x). Then it will be difficult to distinguish the graph from the reduced graph (by replacing two edges $\{y, x\}$, $\{x, z\}$ by an edge of length $d(y, x)+d(x, z)$ joining y and z).

Here we will formulate several version of this problem, mention the relevant results, both new and old, and discuss their algorithmic implications.

First, we give some definitions. For a matrix D with rows and columns indexed by S, we say D has a *realization* if there is a graph whose node set contains S, and $D(u, v)$ is the distance between u and v. It is easy to see that a matrix D has a realization if its entries are nonnegative and satisfy the triangle inequality.

$$D(u, v)+D(v, w) \geqslant D(u, w). \tag{2}$$

It turns out [22] that this necessary condition is also sufficient (as indicated by the example above). We say that D is a *distance matrix* for S if $D(u, v)$, $u, v \in S$, is nonnegative and satisfies (1).

Problem 1. For a given distance matrix D on a set S of terminal nodes, find a graph G which is a realization of D so that the total sum of all edge lengths of G is minimized.

The above problem was first proposed by Hakimi and Yau [22] in 1965 who also gave an algorithm which will lead to the solution for the special case that the realization of the distance matrix is a tree. Since then, an extensive literature has developed for this problem. Special attention has been given to the case of *tree realizations*, i.e., when the graph that realized the distance matrix is a tree. Necessary and sufficient condition for a distance matrix realizable by a tree were given in several papers [4, 16, 26–28]. An $O(n^2)$ time algorithm for testing and constructing a tree realization from a distance matrix was described in [15].

For a (general) distance matrix, it is not too hard to show that an *optimal realization* (i.e., the realization having the minimum total length) exists [17]. It was shown in [17] that an optimal realization can have at most n^4 nodes if the number of terminal nodes is n. On the other hand, there are some examples of optimal realizations of a distance matrix on n terminals which have a least n^2 vertices. Therefore there is a finite (but exponential) algorithm to find an optimal realization for a given distance matrix. Various heuristics are discussed in many papers [7, 25, 22, 28–30]. However, solutions to this problem seem to be elusive, and, in fact, computing optimal realizations for distance matrices with a small number of terminal nodes is already quite complicated. Indeed, Althöfer [3] showed that the problem of finding optimal realizations of distance matrices with integral entries is NP-complete. We will provide here additional evidence pointing to the difficulties in approximating the optimal realization. In Section 3, we will show that there are distance matrices D and D' on n terminal nodes satisfying $D(u, v) \geqslant D'(u, v)$, but where the optimal realization of D' has total edge length much larger than that of D. In fact, the ratio of the respective sums of edge lengths can be as large as a factor of n.

Because the realization problem seems hard even to approximate, we introduce a number of more robust variations and generalizations: For a given distance matrix D on a set S of terminal nodes, a graph G is said to be a *weak* realization of D if

(i) the node set of G contains S,

(ii) the distance between u and v in G is greater than or equal to $D(u, v)$ for all u, v in S.

Problem 2. For a given distance matrix D on a set S of terminal nodes, find a weak realization of D so that the total sum of all edge lengths of G is minimized.

It is not hard to show that if a distance matrix D dominates another matrix D' (i.e., $D(u, v) \geqslant D'(u, v)$), an optimal weak realization of D has total edge length no smaller than that of D'.

For a given distance matrix D on a set S of terminal nodes, a graph G is said to be a *rooted weak* realization of D if

(i) the node set of G contains S,

(ii) the distance between u and v in G is greater than or equal to $D(u, v)$ for all u, v in S,

(iii) there is a special node v^* in S, called the *root*, and the distance between u and v^* in G is equal to $D(u, v)$ for all u, v in S.

Problem 3. For a given distance matrix D on a set S of terminal nodes and a special node v^* of S, find a rooted weak realization of D so that the total sum of all edge lengths of G is minimised.

As it turns out, both Problems 2 and 3 are NP-complete. We will give approximation algorithms for both Problems 2 and 3, which are within a small constant factor of the optimum.

We remark that the weak realization problems are similar to but different from the so-called *Steiner tree problem* which also has an extensive literature [23].

Euclidean Steiner tree problem. Given n points in the plane (or, in general, $\mathbb{R}^n$), find the shortest tree connecting the n points (where this tree may have additional points as vertices).

Graph Steiner tree problem. For a given graph and a subset S of nodes, find the shortest tree containing nodes in S.

Both of the above Steiner problems are NP-complete [19]. These Steiner problems are different from the distance realization problems since the host graph is unknown for the distance realization problems.

There is yet another related version of the realization problem which has more detailed inputs. It is also come up in the applications to Internet tomography:

Problem. Suppose we are given a set S of terminal nodes and a set E of edges. Each pair of terminal nodes, u and v, are associated with a subset of edges which are in a shortest path joining u and v in the host graph. (In other words, an incidence matrix of edges and pairs of nodes is given.) The goal is to determine the host graph (i.e., to describe the adjacencies of its nodes and edges).

The above problem turns out to be a special case of the problem of determining the graphical construction of a matroid. This problem was solved by Tutte in his seminal papers [32, 33]. Indeed, there is a polynomial time algorithm of order $O(n^2m)$ for graphs with m edges and n terminal nodes [6, 18].

4. THE DISTANCE REALIZATION PROBLEM IS HARD TO APPROXIMATE

In this section, we will illustrate the difficulty of approximating the solution for a distance realization problem. We will consider a certain distance matrix M together with another matrix M' such that the corresponding entries of the two matrices have very similar values, but their realizations are vastly different.

THEOREM 1. *For any given positive value ε, there are two distance matrices M and M′ on the same set S of terminal nodes satisfying the following:*

(a) *For all u, v in S,*

$$|M(u, v) - M'(u, v)| \leqslant \varepsilon.$$

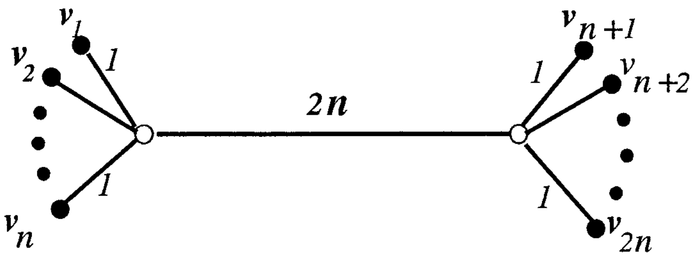

FIG. 2. A tree of total length $4n$.

(b) *Let G denote an optimal realization of M with total length t. Any realization of M′ has length at least* $|S|\ t/8$.

Proof. Let S denote a set of $2n$ terminal nodes, say, $x_1, ..., x_{2n}$. We consider a distance matrix M on S defined by $M(x_i, x_j) = 2$ if $1 \leqslant i < j \leqslant n$ or $n < i < j \leqslant 2n$. Otherwise $M(x_i, x_j) = 2n + 2$.

It is easy to see that M is realized by the tree T as illustrated in Fig. 2. The total length of T is $4n$.

Now, we define M' as follows: For all $i, < j$ we define $M'(x_i, x_j)$ to be a value between $M(x_i, x_j) - \varepsilon$ and $M(x_i, x_j)$ so that values $M'(x_i, x_j)$ are algebraically independent over the rationals.

Let H denote a realization for M'. Let Q denote a set of points whose removal partitions H into two parts, one of which contains $A = \{x_1, ..., x_n\}$ and the other containing $B = S - A$. Here, by a point we mean either a vertex or just a point lying on an edge. We claim that Q contains at least $n/2$ points. Suppose to the contrary that Q contains $r < n/2$ points. We consider all the distances $d(x_i, q)$ in H, for q in Q and $i = 1, ..., 2n$. Since Q is a cutset, any path joining $x_i, x_j, 1 \leqslant i \leqslant n < j$, must contain a point in Q. Therefore the distance $d(x_i, x_j)$ for $x_i, x_j, 1 \leqslant i \leqslant n < j$ in H can be expressed as a sum of $d(x_i, q) + d(x_j, q)$ for some q in Q. There are at most $2nr$ algebraically independent quantities $d(x_i, q)$. However, there are n^2 distances $d(x_i, x_j)$ that are algebraically independent. Therefore, we have $n \leqslant 2r$, which is a contradiction.

We have shown that any cutset contains at least $n/2$ points. Thus, there are at least $n/2$ edge-disjoint paths joining points of A to points B in H. This implies that H has length at least n^2 which is at least $t\ |S|/8$ where t is the length of T. ∎

With almost the identical proof, we have

COROLLARY 1. *There are two distance matrices M and M′ on the same set S of terminal nodes satisfying the following:*

(a) $M(u, v) > M'(u, v)$ *for all u, v in S.*

(b) *Let G denote an optimal realization of M with total length t. Any realization of M′ has length at least* $|S|\ t/4$.

5. THE WEAK REALIZATION PROBLEM

For a given distance matrix D with n terminal nodes, we want to find a shortest graph in which any two terminal nodes u and v have distance at least $D(u, v)$. It is easy to see that a shortest graph which is a weak realization for D must be a tree. Furthermore, if a distance matrix D dominates another distance matrix D', then a realization for D is also a realization for D' and therefore the shortest realization for D is no greater than that for D'. For a set of n terminal nodes, there can be at most $n-2$ Steiner nodes since the weak realization is a tree. So there are finitely many topologies for the weak realization for D. For each topology, the problem of determining a weak realization can be formulated as a linear programming problem. Since there are exponentially many topologies, the brute force approach is impractical when n becomes large. Before we discuss the heuristics for the weak realization problem, we will first prove the following:

THEOREM 2. *The weak realization problem is NP-complete.*

Proof. We will deduce from the NP-complete problem of determining if a graph is 3-chromatic [19].

For a graph G, we first form a graph G' by replacing each vertex v of G by two copies v_1 and v_2 which have the same neighborhood (as the set of copies of neighbors of v). We will say that v_1 and v_2 are *mates* of each other. Also G contains three additional isolated vertices v_1^*, v_2^*, v_3^*. Now, we consider an associated matrix M_G with rows and columns indexed by vertices of G'. We define, for two distinct vertices u and v in G', $M_G(u, v) = 2$ if u and v are adjacent in G', $M_G(u, v) = 0$ if $u = v$, and $M_G(u, v) = 1$ otherwise. Clearly, M_G satisfies the triangle inequality and therefore is a distance matrix.

It suffices to show that a graph G on n vertices has chromatic number 3 if and only if M_G has an optimum weak realization with length $n+9/4$.

First, we will show that there is a weak realization for a 3-chromatic graph G having length at most $n+9/4$. We consider a proper coloring of G in three colors. For each color i, for $i = 1, 2, 3$ we associate a (new) vertex s_i which is adjacent (of length $1/2$) to each vertex (except for v_i^*'s) in color i. There are new Steiner vertices, v^*, u_1, u_2, u_3 so that edges $\{v^*, u_i\}, \{u_i, v_i^*\}, \{u_i, s_i\}$ are all of length $1/4$. It is easy to see that this tree is a weak realization of M_G and its length at most $n+9/4$.

It suffices to show that a 3-chromatic graph on n vertices has a weak realization for M_G with length at least $n+9/4$. We will use induction on n. Let T denote a weak realization of M_G with minimal total length. Either there is a leaf x with its incident edge of length $\geqslant 1$ or there are two leaves adjacent to the same Steiner vertex, one of which must have length at least $1/2$. Therefore there is always a leaf v with its incident edge of length $p \geqslant 1/2$.

Suppose $v \neq v^*$ and the mate of v is not a leaf of the same Steiner point. We can form a new tree T_1 by removing the leaf of v and reconnecting v as a leaf to a point s' of T where s' is not s and s' at distance $1/2$ from the mate of v in T. It is easy to see the resulting tree is still a weak realization of M_G. Therefore we can assume $p = 1/2$. In fact, we can also assume that the mate of v is a leaf of s since otherwise we will consider T_1 instead.

Suppose $v = v_i^*$. We can form a new tree T_2 by removing the leaf of v and reconnecting v as a leaf to a point s'' of T where s'' is not s and s'' at distance $1/2$ from a furthest terminal vertex in T. (Such an s'' exists because of 3-chromaticity). Again, the resulting tree T_2 is a weak realization of M_G. Therefore we may assume v_i^* is not a leaf of s and $p = 1/2$. Also, we conclude that all leaves are of length at most $1/2$. If a Steiner point s has more than one leaf, then all leaves of s are of length exactly $1/2$. We will need the following fact:

Let B_y be the ball consisting of all points on edge segments of distance no more than $1/2$ to a terminal node y in T. Each B_y contains a part of T of length at least $1/2$. Suppose y is a leaf of length $p < 1/2$. Then B_y contains a part of T of length $1-p$. We assign a *weight* of $1/2-p$ to the ball B_y. All balls B_y are disjoint (except for the boundaries). The total length of T is at least $1/2$ times the number of balls plus the weights $w(B_y)$ of all balls. We also observe that the weight of $w(B_y)$ is at least half of the total length of the line segments (except for leaves) of T in B_y. Let T' denote the tree by deleting leaves from T. The total length of T is no smaller than $n+3/2$ plus half of the length of T'. If the length of T' is at least $3/2$, we are done. We may assume that T' has total length less than $3/2$.

We observe that the diameter of T' is at least 1 since there are two terminal nodes in T of distance at least 2 and all leaves are of length $\leqslant 1/2$. We assume that two vertices a and b achieve the maximum distance in T'. That is, $d_{T'}(a, b) = \max_{p,q} d_{T'}(p, q) = 1 + x$ for some $x \geqslant 0$. Let P denote that path joining a and b in T'. A rooted branch B is a connected component of T by deleting edges in P while the root of B is a vertex on P. We partition the set of branches into three parts. The first part consists of all branches with roots at distance less than $1/2$ from a. The second part consists of all branches with roots at distance less than $1/2$ from b. The third part consists of the remaining branches. A terminal node v is said to be incident to the ith part if v is in or adjacent to Steiner points in the ith part. Clearly, all terminal nodes in the first part are of distance less than 2 in T from each other. Similarly, all terminal nodes incident to the second part are of distance less than 2 in T from each other. Any terminal node v in or adjacent to Steiner points in the third part are of distance less than $1-x$ to P and less than $3/2$ from a or b in T. Therefore v is at distance less than 2 from all other terminal nodes. This implies that G can be properly colored by two colors, which is a contradiction. Therefore we conclude that T' must have length at least $3/2$. Therefore we have proved that if G has chromatic number 3, then M_G has a weak realization of length $n+9/4$.

It remains to show that if M_G has a weak realization T of length $n+9/4$, then G has chromatic number at most 3. If the diameter of T is more than 2, then the preceeding arguments still work and G is bipartite. We may assume that T has diameter 2 and we can assume that $d_T(a, b) = 2$. The vertices of T can be partitioned into three parts, as above. It is easy to see that in each part the terminal nodes are within distance less than 2 from one another in T. Therefore the terminal nodes of each part are nonadjacent to each other in G. Therefore G has chromatic number 3.

This completes the proof of NP-completeness of the weak realization problem.

Heuristics for Realization

Here we consider some of the heuristic algorithms for the weak realization problem.

ALGORITHM A. For a set of n terminal nodes and the associated matrix M, first construct the minimum spanning tree T_0 using M. Then, for any two incident edges e and e' in the current tree T_i, we take a local optimization step if there is no critical path passing through e and e'. Namely, we replace e, e' by three edges e_1, e_2, e_3 all incident to a new Steiner point w satisfying:

(a) $w(e_1)+w(e_3)=w(e), w(e_3)+w(e_3)=w(e')$.

(b) Choose $w(e_3)$ as large as possible so that the distances between terminal nodes in the tree dominate the distances given in M. Thus, either e_1 and e_2 are in a critical path or one of e_1 or e_2 has zero length.

(c) Remove any Steiner vertex with degree 2 (using a long edge instead). Repeat the above local optimization step until the graph is stable (i.e., no further local optimization can shorten the total length.)

Unlike the case for the strong realization problem, we will show that the above algorithm yields a tree with a length of within a factor of 2 of the optimum. In fact, this is true without making any local optimization steps.

LEMMA 1. *A minimum spanning tree of a distance matrix M is a weak realization of M having length within a factor of $2-2/n$ of the optimum solution for the weak realization problem.*

Proof. Suppose T is an optimum weak realization of M. We note that twice the length of T is the sum of the length of edges in a Hamiltonian cycle C in G. Suppose we delete the longest edge in C. Its length is at least as large as that of the minimum spanning tree. The lemma then follows. ∎

It would be of interest to know if the local optimization steps above actually help reduce the upper bound with the factor of $2-2/n$. Unfortunately, the answer is negative as shown by the following example.

EXAMPLE. We consider the distance matrix M for a set of $4n$ terminal vertices, say $v_1, v_2, \ldots, v_{4n}$, defined as follows: $M(v_i, v_{i+2})=2$ for $i=1, \ldots, 4n-1$ and all other distances have value 1.

It is easy to see that a minimum spanning tree is a path on $4n$ vertices with total length $4n-1$ as shown in Fig. 3. In fact, Algorithm A will generate such a path. However, M has the following weak realization which has total length $2n+1$ as shown in Fig. 4.

FIG. 3. A path of length $4n$.

LEMMA 2. *If a distance matrix M has a strong tree realization T, then T is the unique minimum length weak realization.*

Proof. Let T' be a weak realization. We will show that if T' is not T, T' is not a minimum length weak realization and therefore, T must be the unique minimum length weak realization. If T' is not a tree, it is not minimum length since we can remove an edge from a cycle. If T' has any Steiner points as leaves, we can remove them. So T' has no Steiner leaves and T' is a tree.

If T' is not a strong realization, there must be some pair of points v, w, so that $M(v, w)$ is less than the length of the path joining v and w in T'. Now embed T' in the plane so that the path joining v, w is on the x-axis, and the rest of the points are above the x-axis, so that no edges cross. Number all the regular vertices in T' in order by traversing from v to w, say, $v, v_2, v_3, \ldots, v_{k-1}, w$. Consider the walk W that visits these nodes in this order and then returns to v. It visits each arc twice, so $l(W) = 2l(T')$. It is made up of a sequence of shortest paths in T', each of which is at least as large as a corresponding entry in M and one of which is larger than a particular entry in M: So we have

$$2 * l(T') > M(v, v_2) + M(v_2, v_3) + \cdots + M(v_{k-1}, w) + M(w, v).$$

This sum, in turn, must equal the length of the corresponding walk in T, because the path lengths in T must equal the matrix entries. This walk in T visits each leaf of T, so it must visit each edge at least twice. Therefore, we have $2 * l(T') > 2 * l(T)$. Hence T' is not minimum length. We have proved that if T' is a strong realization, then $T' = T$. ▮

Here we consider another heuristic which can be used to improve the solution generated by Algorithm A.

ALGORITHM 1: *Opt.* For a set of n terminal nodes and an associated matrix M, first run Algorithm A and reach a stable tree T. Modify the tree by the following steps.

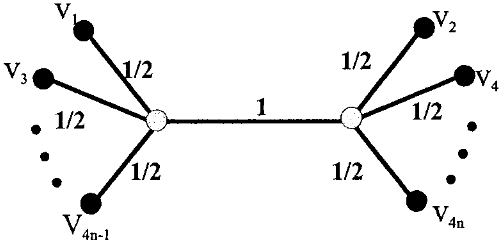

FIG. 4. A weak realization of length $2n+1$.

(i) Choose an edge e of T and consider $T-e$.

(ii) Add a (random) edge joining a pair of terminal nodes u and v with edge weight $M(u, v)$ to form a tree T'.

(iii) If the resulting tree T' is not stable, repeat the local optimization steps as in Algorithm A until we reach a stable tree T''. If T'' has length shorter than T, then we *accept* this change. Otherwise, try another 1-opt step on T. Continue, until no 1-opt step leads to an improvement. This result in a *1-opt stable* tree.

Conceivably, the 1-opt algorithm can further improve the resulting trees. However, we are not able to derive bounds for such an improvement.

We remark that a natural generalization is to have an analogous 2-opt algorithm or even more general algorithm of this type.

6. THE ROOTED WEAK REALIZATION PROBLEM

For a given distance matrix D with n terminal nodes including a spectral node v^*, called the root, we want to find a shortest graph in which any two terminal nodes u and v have distance at least $D(u, v)$ and, in particular, the distance of any vertex u from v^* is exactly $D(u, v^*)$. In other words, the above problem is to find the weak realization which realizes the distances from a special node. This problem can be related to the distance realization problem as follows:

Suppose we start with a known host graph H. For a set S of terminal nodes, let $D_H(u, v)$, for u, v in S, denote the distance between u and v in H. For the matrix D_H, and a fixed vertex as the root, a breadth-first tree is a rooted weak realization of D_H.

Suppose the host graph is unknown. We are given a distance matrix D_H and the problem of interest is to recover the host graph realizing D_H. Suppose we have all rooted weak realizations T_v ranging over all terminal nodes as the roots v. The host graph H obviously contains all T_v as subgraphs of H. There is a large literature on the problem of finding *universal graphs* which contain a given family of graphs as subgraphs [2, 9–11]. Although the problem of determining the optimum universal graph is again a hard problem, it is worth mentioning this line of approach which could be feasible for special applications.

In the remainder of this section, we focus on the rooted weak realization problem.

THEOREM 3. *The rooted weak realization problem is NP-complete.*

Proof. The proof is quite similar to that of Theorem 2. The reduction is again from the problem of determining the chromatic number of a graph. ∎

For a graph G and a root v^*, we consider an associated graph M_G with rows and columns indexed by vertices of G. We define $M_G(v^*, u) = 1$ for all u and for two vertices u and v which are not v^*, we define $M_G(u, v) = 2$ if u and v are adjacent and $M_G(u, v) = 1$ otherwise. Clearly, M_G satisfies the triangle inequality and is therefore a distance matrix.

It suffices to show that a graph G on n vertices has chromatic number χ if and only if M_G has a weak realization with length $(n+\chi-1)/2$. The proof is essentially the same as that in the previous theorem and will be omitted. ▮

7. ALGORITHM IMPLEMENTATIONS

A research prototype monitoring systems (named FELIX) that was implemented at Telcordia Technologies [20] concerns the whole range of issues concerning monitor design, data collection and storage, Web-based graphical user interface, and survivability analysis. Here we only focus on the module involving the implementation of the distance realization algorithms. In Fig. 5, the diagrams show the sequence of software components, intermediate files, and the Web-based GUI (graphical user interface) that controls the overall operations. The surrounding systems allow a user to use data for a distance matrix from a realistic Internet graph or a random generated Internet graph (using models in [1, 8]) with selected parameters (which are controlled and executed in "*fan.cgi*" and "*real*"). The reduced graph is generated (through "*tier*" and "*reduce*." The main software modules generate strong realizations (by "*realize*"), weak realizations (by "*weak*") and rooted weak realizations (by "*weak-rall*" that are then merged into a realization (by "*tree-merge*"). The topology files (before and after discovery) can be viewed and compared.

The strong realization heuristic was based on the methods described in [22]. It started with a complete graph on the terminal nodes, where the length of an edge from i to j was the matrix entry m_{ij}. It then consisted of an initial phase of replacing each triangle found of three edges e_1, e_2, e_3, by an equivalent "Y" of three edges e'_1, e'_2, e'_3 and a Steiner point (and half the total length of the triangle). No special order was chosen for the triangles. This was followed by alternating phases of the local optimization steps described in Algorithm A (which works for strong realization as well as weak ones) and of steps of removing redundant edges. The local optimization steps proceeded through the nodes in each order, and at each node choosing the optimization steps in order of greatest savings of length, until no

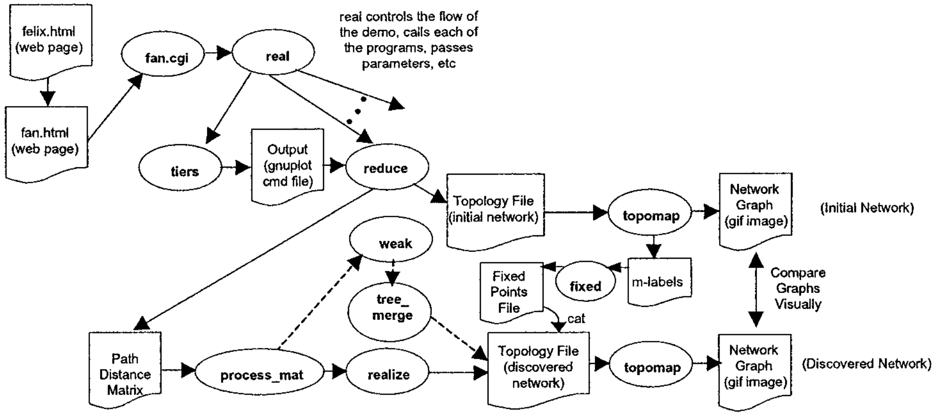

FIG. 5. A flow chart of distance realization method for inferencing network topology.

improvement could be made. The removal of redundant edges proceeded through the edges in order of decreasing length, removing each edge that could be removed while keeping the required distances. These two phases alternated until no further changes occurred.

The weak realization heuristic was an implementation of Algorithm 1-Opt above. As for the strong realization heuristic, the simple local optimization steps were done proceeding through the nodes, at each node choosing the local optimizations greedily. The 1-opt steps examined each edge e in order and checked each pair of terminals, one from each of the two components of $T-e$. There was no randomization.

The rooted weak realization heuristic was a modification of Algorithm 1-Opt. Instead of a minimum spanning tree, the starting graph was a star centered on the root. In the 1-opt step, unlike for normal weak realizations, if we delete an edge e from the current tree T and replace it by another edge between the resulting two components, only for certain positions of this new edge can we obtain a weak rooted realization without changing lengths of the other edges. We kept the end of e away from the root fixed and only considered as candidates for the other end of e nodes so that we could get such a realization and were either leaves or had at least one neighbor where we could not get such realization.

We implemented two algorithms for merging rooted weak realizations into a strong realization. The first algorithm was based on the strong realization algorithm. The starting graph was the union of the weak realizations, with the various copies from each weak realization of each terminal merged, but all Steiner nodes kept distinct. The local optimization steps started off with a phase of only attempting to merge two edges if they came from different rooted weak realizations. This should preserve the property of being a universal graph for the weak rooted realizations. Furthermore, at this stage, if the rooted weak realizations were locally minimal, any attempt to merge two edges from the same rooted weak realization should fail. In the sparser examples this produced a locally minimal or nearly minimal strong realization, but in the denser examples this left the graph with redundant shortest paths between terminals. (Here "sparser" and "denser" refer to the simulated Internet graphs for the net test problems.) For further reduction therefore, this was followed by alternating phases of deleting redundant edges and more local optimization steps, stopping when no further changes occur.

We also tested a second strong realization heuristic, which used a rooted weak realization as a starting point. First, choose a root v_0, and find a locally optimal rooted weak realization using the rooted weak realization algorithm described above. Then for each remaining terminal v_i in sequence, connect it by an edge (v_i, v_j) of length equal to the specified distance $m_i j$ to every other terminal v_j not already at this specified distance $m_i j$. For this terminal v_i, apply the local improvements of the strong realization algorithm to the resulting graph with the caveat that distances between terminals v_j and v_k, where $i < j$ and $i < k$, are allowed to exceed $m_j k$. When this has been done for the last terminal, the resulting graph will be a strong realization (Fig. 6).

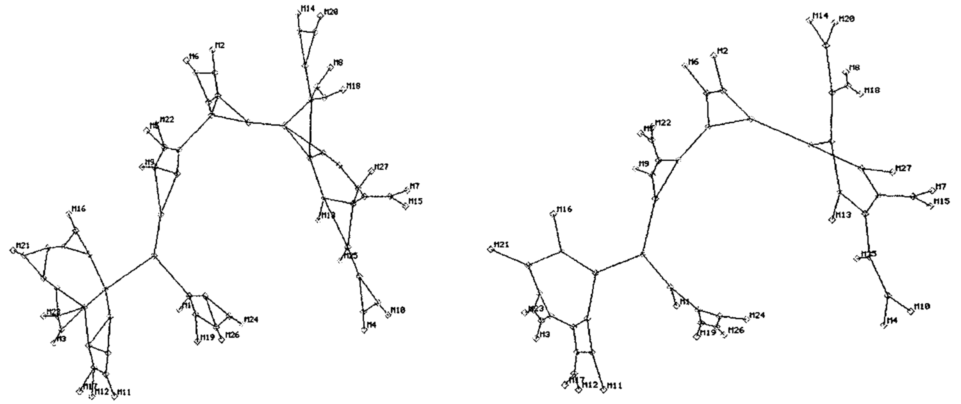

FIG. 6. An example of original network graph and the corresponding graph discovered by the distance realization method.

Clearly, there is room for further refinement of heuristics and improvement algorithms for realization problems derived from distance relay measurements.

REFERENCES

1. W. Aiello, F. Chung, and L. Lu, A random graph model for massive graphs, *in* "Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing," pp. 171–180, 2000.
2. N. Alon, M. Capalbo, Y. Kohayakawa, V. Rödl, A. Rucinski, and E. Szemer'edi, Universality and tolerance, *in* "The 41st Annual Symposium on Foundations of Computer Science," 2000.
3. L. Althöfer, On optimal realizations of finite metric spaces by graphs, *Discrete Comput. Geom.* **3** (1988), 103–122.
4. H. Bandelt, Recognition of tree metrics, *SIAM J. Discrete Math.* **3** (1990), 1–6.
5. K. Bharat and A. Broder, A technique for measuring the relative size and overlap of public web search engines, *in* "Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia," pp. 379–388, 1998.
6. R. E. Bixby and D. K. Wagner, An almost linear-time algorithm for graph realization, *Math. Oper. Res.* **13** (1988), 99–123.
7. F. T. Boesch, Properties of the distance matrix of a tree, *Quart. Appl. Math.* **26** (1968/69), 607–609.
8. K. Calvert, M. Doar, and E. W. Zegura, Modeling Internet Topology, *IEEE Comm. Mag.* **35** (1997), 160–163.
9. F. R. K. Chung and R. L. Graham, On graphs which contain all small trees, *J. Combin. Theory (B)* **24** (1978), 14–23.
10. F. R. K. Chung and R. L. Graham, On universal graphs, *Ann. New York Acad. Sci.* **319** (1979), 136–140.
11. F. R. K. Chung and R. L. Graham, On universal graphs for spanning trees, *J. London Math. Soc.* **27** (1983), 203–211.
12. F. R. K. Chung, Universal graphs and induced universal graphs, *J. Graph Theory* **14** (1990), 443–454.
13. K. C. Claffy, T. E. Monk, and D. McRobb, Internet tomography, Nature, *Web Matters*, available to http://helix.nature.com/webmatters/tomog.html, Jan. 1999.
14. K. C. Claffy and T. Monk, What's next for Internet data analysis? *IEEE Special Issue Comm. 21st Century* **85** (1997), 1563–1571.

15. J. C. Culberson and P. Rudnicki, A fast algorithm for constructing trees from distance matrices, *Inform. Process. Lett.* **30** (1989), 215–220.

16. E. Dahlhaus, Fast parallel recognition of ultrametrics and tree metrics, *SIAM J. Discrete Math.* **6** (1993), 523–532.

17. A. W. M. Dress, Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces, *Adv. in Math.* **53** (1984), 321–402.

18. S. Fujishige, An efficient PQ-graph algorithm for solving the graph realization problem, *J. Comput. System Sci.* **21** (1980), 63–86.

19. M. R. Garey and D. S. Johnson, "Computers and Intractability, A Guide to the Theory of NP-completeness," W. H. Freeman, San Francisco, 1979.

20. B. Siegell, J. DesMarais, M. Garrett, P. Seymour, and D. Shallcross, "Felix Project: Topology Discovery from One-way Delay Measurements," Waikato Workshop on Passive and Active Measurement, Waikato, New Zealand, April 2000.

21. M. Goemans and D. Williamson, Improved approximation algorithms for maximum cut and satisfiability problem using semidefinite programming, *J. Assoc. Mach.* **42** (1995), 1115–1145.

22. S. L. Hakimi and S. S. Yau, Distance matrix of a graph and its realizability, *Quart. Appl. Math.* **22** (1965), 305–317.

23. F. K. Hwang, D. S. Richard, and P. Winter, The Steiner problem, *Ann. Discrete Math.* **53** (1992).

24. T. Munzner, Exploring large graphs in 3D hyperbolic space, *IEEE Comput. Graphics Appl.* **18** (1998), 18–23.

25. J. Nieminen, Realizing the distance matrix of a graph, *Elektron. Inform. Kybernet.* **12** (1976), 29–31.

26. A. N. Patrinos and S. L. Hakimi, The distance matrix of a graph and its realization, *Quart. Appl. Math.* **30** (1972/73), 255–269.

27. J. M. S. Simoes Pereira, A note on the tree realizability of a distance matrix, *J. Combin. Theory* **6** (1969), 303–310.

28. J. M. S. Simoes Pereira, An algorithm and its role in the study of optimal graph realizations of distance matrices, *Discrete Math.* **79** (1990), 299–312.

29. J. M. S. Simoes Pereira, An optimality criterion for graph embeddings of metrics, *SIAM J. Discrete Math.* **1** (1988), 223–229.

30. J. M. S. Simoes Pereira and C. M. Zamfirescu, Submatrices of non-tree-realizable distance matrices, *Linear Algebra Appl.* **44** (1982), 1–17.

31. I. Peterson, Diagnosing the Internet's ills, *Science News* **155** (1999), 37.

32. W. T. Tutte, A homotopy theorem for matroids, I, II, *Trans. Amer. Math. Soc.* **88** (1958), 144–174.

33. W. T. Tutte, An algorithm for determining whether a given binary matroid is graphic, *Proc. Amer. Math. Soc.* **11** (1960), 905–917.