Unlikelihood That Minimal Phylogenies for a Realistic Biological Study Can Be Constructed in Reasonable Computational Time

R. L. GRAHAM
Bell Laboratories, Murray Hill, New Jersey

AND

L. R. FOULDS
Operations Research, University of Canterbury, New Zealand

Received 11 September 1981; revised 13 January 1982

ABSTRACT

The problem of determining a phylogeny of maximum parsimony from a given set of protein sequences is defined. It is shown that this problem is what is called, in computer science, NP-complete. The implication of this result is that it is equivalent in difficulty to a host of other problems in combinatorial optimization which are notorious for their intractability. This implies that it is more fruitful to attempt to develop heuristic techniques (which do not guarantee maximum parsimony but which do run in reasonable computer time) than to try to develop exact algorithms for phylogeny construction.

1. INTRODUCTION

There have been many recent attempts by scientists to analyze the variance between proteins of different taxa in order to produce phylogenetic trees or phylogenies of maximum parsimony. (See for example Fitch and Margoliash [4], Foulds et al. [8], and Hendy et al. [15].) This paper examines the computational complexity of a specific variation of this problem. A phylogeny is to be constructed for a given set of taxa by using a set of homologous sequences derived from the same protein, one for each taxon, that are all of the same length. We shall deal with nucleotide sequences, and assume that each character in each sequence will be occupied by exactly one of the four character states: A, C, G, or U. (Amino acid sequences can be readily converted into nucleotide sequences by any of a number of methods, including those of Fitch and Margoliash [4], Moore et al. [16], and Penny et al. [17].)

The optimality criterion is to minimize the number of nucleotide substitutions. It is equivalent to the minimum mutation distance criterion of Moore et al. [16] and includes the total of the minimum weight of the amino acid

52 Vanderbilt Ave., New York, NY 10017 0025-5564/82/06133+10$02.75

changes together with any additional nucleotide substitutions that are necessary because of the structure of the phylogeny and sequence redundancy. Use of this criterion has been justified by Fitch [7] and others. A method for calculating the minimum possible number of nucleotide substitutions for a given phylogeny was presented by Fitch [5] and by Fitch and Farris [6]. The method was placed on a firm mathematical basis by Hartigan [14] and by Moore et al. [16].

It is assumed that nucleotide substitution is reversible in the sense that if nucleotide α is substituted for nucleotide β at a particular character of a specific sequence, then it is possible for the later substitution $j: \beta \to \alpha$ to occur. This assumption represents a generalization of the problem of Camin and Sokal [1] and of Eastabrook [3], where substitutions were considered irreversible. Thus the point in the phylogeny representing the common ancestor of all species does not affect the minimality criterion, and the phylogeny produced will have undirected branches or links.

The problem is to construct a phylogeny in which each operational taxonomic unit (OTU) is represented by a unique point. All other points represent hypothetical taxonomic units (HTUs), each with a new, distinct sequence introduced in order to minimize the total number of nucleotide substitutions required to connect the given set of taxa. Note that it is *not* necessary for the OTUs to be associated with only the branch tips or end points of the phylogeny, nor for the phylogeny to be bifurcating in the sense that all interior points are of degree three. This represents a departure from the more restrictive problem studied by Fitch [7].

It is assumed that all sequences are contracted by the removal of their character states at character j if j is a singularity or an equivalent position, as defined in Fitch [7]. Thus only positions which can affect the structure of the most parsimonious phylogeny are present.

We turn now to identification of the most parsimonious phylogeny. Let N be the common sequence length and M be the number of taxa under study. As N and M are finite, it would appear theoretically possible to enumerate all possible phylogenies and choose those with the minimum number of substitutions. Unfortunately N and M are usually at least 100 and 20 respectively in any realistic biological study. This means that the number of possibilities is enormous. The actual number of phylogenies for specific values of M has been calculated for both general and bifurcating trees by Foulds and Robinson [9, 10]. In order to establish the complexity of the problem we develop some mathematical machinery in the next section.

2. THE STEINER PROBLEM IN PHYLOGENY

A mathematical definition of the problem described in the last section will now be developed with the use of graph theory. A *graph*[1] G comprises an

[1] For undefined graph-theoretic terminology see Harary [13].

ordered pair of sets (V, E) where V is a nonempty, finite set of *points* and E is a set of unordered pairs of points of V, called *lines*. Each line $e \in E$ is commonly denoted by the form $\{v_i, v_j\}$, the unordered pair of points $v_i, v_j \in V$ which it represents. A *weighted graph* comprises an ordered pair (G, w) where G is a graph and w is a function $w: E \to \mathbb{R}$ where E is the line set of G. It is common to denote $w(\{v_i, v_j\})$ by w_{ij} for each $\{v_i, v_j\} \in E$. A *path* between points v_0 and v_m in a graph $G = (V, E)$ is a sequence of the form

$$\langle v_0, \{v_0, v_1\}, v_1, \{v_1, v_2\}, v_2, \ldots, \{v_{m-1}, v_m\}, v_m \rangle$$

where $v_i \in V$, and $\{v_i, v_{i+1}\} \in E$ for $i = 0, 1, 2, \ldots, m-1$. If $v_0 = v_m$, the resulting structure is termed a *cycle*. A graph without cycles is termed *acyclic*.

If $S \subseteq V$, S is said to be *connected in* G if there exists at least one path in G between every pair of points in S. If V is connected in G, then the graph $G = (V, E)$ is said to be *connected*.

The *Steiner problem in graphs* (SPG) involve a connected weighted graph (G, w) where $G = (V, E)$ and a nonempty set $X \subseteq V$. The objective is to find a subset $T \subseteq E$ such that

(1) (X, T) is a tree (a connected acyclic graph) called a Steiner minimal tree (SMT),

(2) $\Sigma_{\{p_i, p_j\} \in T} w_{ij}$ is a minimum.

The optimization problem described in the introduction is a close cousin of the SPG, and we now show the relationship.

Let A be the set of symbols called *characters* of which the sequences are made up. Let $|A| = r$. Let $S = \{s_1, s_2, \ldots, s_M\}$ be the given set of M sequences. Then if each sequence is N characters long, $S \subseteq A \times A \times \cdots \times A = A^N$. Let f_{ij} = the jth character of s_i, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$.

We now define a *distance function* $d: A^N \times A^N \to \mathbb{N}$. For $\bar{s}_u, \bar{s}_v \in A^N$ define

$$\delta_j^{uv} = \begin{cases} 1 & \text{if } f_{uj} \neq f_{vj} \\ 0 & \text{otherwise} \end{cases}, \qquad j = 1, 2, \ldots, N.$$

Define

$$d(\bar{s}_u, \bar{s}_v) = \sum_{j=1}^{N} \delta_j^{uv}.$$

That is, $d(\bar{s}_u, \bar{s}_v)$ is the number of positions at which $\bar{s}_u$ and $\bar{s}_v$ differ.

Consider now the weighted graph $(\bar{G}, d)$ where $\bar{G} = (A^N, S^*)$ and $S^* = \{\{\bar{s}_u, \bar{s}_v\} : d(\bar{s}_u, \bar{s}_v) = 1\}$. The problem of constructing the phylogeny of maximum parsimony is a special case of the SPG on $\bar{G}$ where X, the given subset of points to be connected, is S. We call this problem the *Steiner problem in phylogeny* (SPP).

In the next section we introduce a special class of problems, called NP-complete, to which the SPP belongs.

3. THE COMPLEXITY OF COMBINATORIAL OPTIMIZATION PROBLEMS

We begin this section by introducing the subject of *combinatorial optimization* (CO), as the SPP is an example of a CO problem. *Combinatorial mathematics* is often described as the study of the arrangement and selection of discrete objects. A part of this subject, *combinatorial optimization*, is concerned with identifying the best possible arrangement or selection from among all those possible. As there are usually a finite number of possibilities for any given problem, it is theoretically possible to examine them all and choose the best. Unfortunately, for most nontrivial problems there are too many solutions for this approach to be feasible. For example, consider the problem finding the shortest path which visits each of a given set of n cities. Even if each of the $n!$ possible paths could be evaluated in a billionth of a second, it would still take over 16,000 years to find the best when $n = 21$.

It is therefore of interest to attempt to design algorithms which are more effective than complete enumeration. We turn now to the question of evaluating the effectiveness of an algorithm. The concept of effectiveness was placed on a firm scientific foundation by Jack Edmonds [2], whose work caused the following convention to be adopted generally by those concerned with algorithm efficiency:

An algorithm is considered to be *effective* if it can guarantee to solve any instance of the problem for which it was designed by performing a number of elementary computational steps and the number can be bounded by a polynomial function of the size of the problem.

It is assumed that computational time is linearly proportional to the number of elementary computational steps required to implement the algorithm. The *size* of a specific instance of a problem is defined to be the number of symbols required to describe it.

It is a valid question to ask whether an effective, or "polynomially bounded time," algorithm can be devised for a given CO problem. There exist CO problems for which it has been shown that no effective algorithm exists, and others for which polynomial time algorithms have been devised. This second class of problems is denoted by P (for polynomial). However, one important point should be noted. For certain instances of practical problem size, say m, there are problems in P that are intractable and problems not in P that are tractable. For example, if $m = 1000$ a problem not in P with an algorithm requiring time of order $e^{0.001m}$ is tractable. But a problem in P with an algorithm requiring time of order m^9 $(= 10^{27})$ is intractable. Some mathematical programming problems have the former

character, and some tensor inversion problems have the latter character. The comparison of efficiency of the simplex and ellipsoidal algorithms for the linear programming problem further brings out this point.

There exists a third class of problems, each of whose status is unknown. It is possible to devise algorithms for each problem, but no effective algorithm is known yet for the class; neither has there appeared a proof showing that it is intractable. Our problem of finding the shortest path through a given set of cities lies in this last class which is denoted by NP (for nondeterministic polynomial). Within NP there is a subset of problems which are called NP-complete. A problem is termed NP-complete if it (1) belongs to NP and (2) has the property that if an effective algorithm is found for it, then an effective algorithm can be found for every problem in NP. In this sense the NP-complete problems are the hardest in NP. To establish the status of a CO problem for which no effective algorithm is known, it is usual to employ the concept of *reducibility*. A problem p_1 is said to be reducible to the problem p_2 (written $p_1 \propto p_2$) $\Leftrightarrow$ (the existence of an effective algorithm for p_2 $\Rightarrow$ the existence of an effective algorithm for p_1). The following result is often used to establish that a problem $p_2 \in$ NP is NP-complete.

THEOREM 3.1

If p_1 is NP*-complete and $p_1 \propto p_2$, then p_2 is also* NP*-complete.*

Proof. Garey and Johnson [1, p. 38].

There is another relevant result.

THEOREM 3.2

If there exists p, an NP*-complete problem for which an effective algorithm exists, then* P = NP.

Proof. Garey and Johnson [1].

Many of the problems in NP have defied the attempts to find effective algorithms of some of the best mathematicians over the past 30 years. There is also more objective circumstantial evidence that P $\neq$ NP. Thus it seems unlikely that an effective algorithm exists for any of the NP-complete problems. Hence the fact that a problem is NP-complete is considered justification for heuristic procedures to be applied to it, that is, procedures which do not guarantee to produce an optimal solution for every instance of the problem. The challenge is to find heuristics with good performance guarantees which are also effective in the sense defined earlier.

What we shall show in the next section is that even when A comprises just two elements, the SPP for A^N is NP-complete.[2]

---

[2]Strictly speaking, we really should be considering the problem of deciding whether X has a Steiner tree with length at most some prespecified value L.

4. THE SPP IS NP-COMPLETE

THEOREM 4.1 (THE MAIN RESULT)

Let $A=\{0,1\}$. *For a fixed positive integer* N, *denote* A^N *by* Q_N. *The weighted graph* $(\bar{G}, d)$ *with point set* Q_N *is just the* 1-*skeleton of the* N-*cube. To show that the Steiner problem for* Q_N *is* NP-*complete* (*which we will denote by* SPQ), *we will reduce the known* [12] NP-*Complete problem of the exact* 3-*cover to* SPQ. *A general instance of the exact* 3-*cover* (X3C) *has the following form*:

Input: $\mathcal{H}=\{F_1, F_2, \ldots, F_n\}$, *where* $|F_i|=3$ *and* $F_i \subseteq \{1,2,\ldots,3m\} \equiv I_{3m}$, $1 \leq i \leq n$;

X3C: *Does* $\mathcal{H}$ *contain* m *sets* $F_{i_1}, \ldots, F_{i_m}$ *whose union is* I_{3m}?

Note that if $\mathcal{H}$ does contain such F_{i_k}, then they must be disjoint.

We now give the details for the construction of the desired corresponding instance of SPQ. To begin with we set $N=4m(n+3m+1)$. A point $q=(q_1,\ldots,q_N)\in Q_N$ can be thought of as consisting of $n+3m+1$ blocks, each of length $4m$:

$$q=(X_0, X_1, \ldots, X_{3m}; Y_1, \ldots, Y_n). \tag{1}$$

To each integer i, $0 \leq i \leq 3m$, define a point x_i by taking in (1)

$$X_i = \left(\overbrace{1,1,\ldots,1}^{4m}\right) \equiv \bar{1}$$

and all other X_j and all Y_k to be

$$\left(\overbrace{0,0,\ldots,0}^{4m}\right) \equiv \bar{0}.$$

Similarly, for $1 \leq j \leq n$, define s_j to have $X_i=\bar{0}$ for all i, $Y_j=\bar{1}$ and $Y_k=\bar{0}$ for all $k \neq j$. Intuitively, for $i \neq 0$, x_i will correspond to the integer i, and s_j will correspond to the 3-set F_j.

Next, if $i \in F_j$ we define a sequence of points $x_{i,j}(k)$, $0 \leq k \leq 8m-1$, as follows:

$$x_{i,j}(k)=(X_0,\ldots,X_i(k),\ldots,X_{3m}; Y_1,\ldots,Y_j(k),\ldots,Y_n)$$

where $X_u = \bar{0}$, $u \neq i$, $Y_v = \bar{0}$, $v \neq j$, and

$$X_i = \bar{1},$$

$$Y_j^{(k)} = \left(\overbrace{0,0,\ldots,0}^{4m-k}, \overbrace{1,1,\ldots,1}^{k}\right), \qquad 0 \leq k \leq 4m,$$

$$X_i^{(k)} = \left(\overbrace{0,0,\ldots,0}^{k-4m}\overbrace{1,1,\ldots,1}^{8m-k}\right), \qquad 4m < k \leq 8m-1,$$

$$Y_j = \bar{1}.$$

Also define $x_{0,j}(k)$ as above for all j, k where $1 \leq j \leq n$, $0 \leq k \leq 8m-1$.

Note that $x_{i,j}(0) = x_i$ for $1 \leq i \leq 3m$. Observe (for future reference) that the $x_{i,j}(k)$ form "chains" from x_i to s_j where consecutive points on the chain have distance 1.

The set $X = X(\mathcal{H})$ will consist of the $8m(3m+1)n$ points $\{x_{i,j}(k): 0 \leq i \leq 3m,\ 1 \leq j \leq n,\ 0 \leq k \leq 8m-1\}$. We point out that X has a *spanning* tree with maximum line weight 2. This implies (see [11]) that any line in an SMT for X has length at most 2. Define

$$L_0 = 4_n(8m-1)+4m,$$

and let $L_S(X)$ denote the weight of an SMT for X.

THEOREM 4.2

If $\mathcal{H}$ has an X3C *then* $L_S(X) \leq L_0$.

Proof Let $F_{j_1}, \ldots, F_{j_m}$ be an exact 3-cover of I_{3m}. For $1 \leq k \leq m$, adjoin to X the Steiner points $s_k = (X_0, \ldots, X_{3m}, Y_1, \ldots, Y_n)$ with $X_i = \bar{0}$, $1 \leq i \leq 3m$, $Y_{j_k} = \bar{1}$, $Y_j = \bar{0}$, $j \neq j_k$. An easy calculation shows that $X^+ = X \cup \{s_k: 1 \leq k \leq m\}$ has a spanning tree of weight L_0; just form the spanning tree consisting of all weight 1 lines for X^+.

Let us call an SMT for X *greedy* if it uses all the weight 1 lines between points in X.

THEOREM 4.3

X has a greedy SMT with length $L_S(X)$.

Proof. Suppose T is an SMT for X with weight strictly less than any greedy SMT. Thus, some line e of weight 1 does not occur in T. Adjoin e to T, thereby forming a cycle C. Some line e' in C must be incident to a Steiner point s of T (since the weight 1 lines in X do not form any cycles). Form the SMT T' by deleting the line e'. Since

$$\text{weight}(e') \geq 1 = \text{weight}(e),$$

then

$$\text{weight}(T') \leqslant \text{weight}(T).$$

However, T is by hypotheses an SMT, so that in fact

$$\text{weight}(L') = \text{weight}(T) = L_S(X).$$

Note that T' contains one more weight 1 line between points of X than T. Theorem 4.3 now follows by induction.

THEOREM 4.4

If $L_S(X) \leqslant L_0$ *then* $\mathcal{H}$ *has an* X3C.

Proof. By Theorem 4.3 we can assume X has a greedy SMT T' with $\text{weight}(T') \leqslant L_0$. As usual, we can assume without loss of generality that every Steiner point of T' has degree at least 3. For $0 \leqslant i \leqslant 3m$, define the subtree T_i of T' to be the tree induced by the points $x_{i,j}(k)$, $1 \leqslant j \leqslant n$, $0 \leqslant k \leqslant 8m-1$. By construction, all lines of T_i have weight 1. We can think of T' as being formed by connecting the T_i together with a set E of lines, each of which is incident to some Steiner point. Observe that since

$$\sum_i \text{weight}(T_i) = 4n(8m-1), \tag{2}$$

then

$$\sum_{e \in E} \text{weight}(e) \leqslant 4m. \tag{3}$$

A key fact to be noted is this: For any $i_1 < i_2 < i_3 < i_4 < i_5$ and any $q \in Q_N$,

$$\sum_{k=1}^{5} d(q, T_{i_k}) > 4m, \tag{4}$$

where $d(q, T_i)$ denotes the minimum distance from q to a point of T_i. [This is the reason that subblocks of length $4m$ are used in the definition of the $x_{i,j}(k)$.]

A consequence of this observation is that *no component of E can have more than two Steiner points*. Otherwise, some Steiner point of T' would be connected by paths in E to (at least) five different T_i's which, by (3), would force

$$\text{weight}(T') > 4n(8m-1) + 4m = L_0.$$

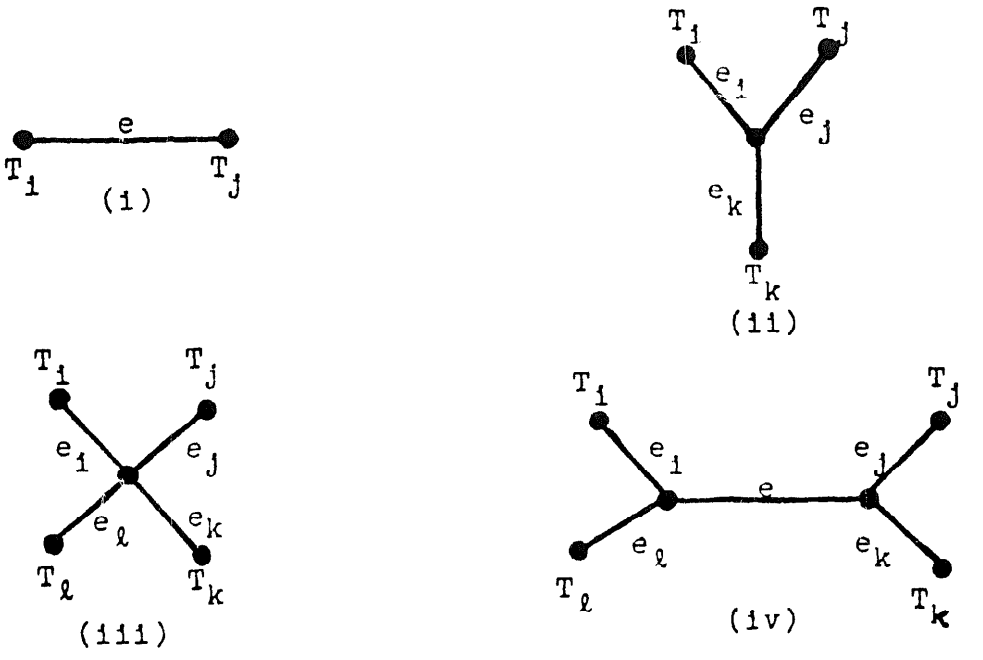

FIG. 1. Possible components of E_k

Thus, there are at most four types of connected components E_k which can be formed by lines from E (also called full Steiner subtrees; see [11]). They are shown in Figure 1.

In Table 1 we list for each case a lower bound on the weight of E_k, the *decrease* $\Delta(E_k)$ in the number of components due to E_k, and ρ, a lower bound on the ratio weight$(E_k)/\Delta(E_k)$. Note that for all of E, weight$(E) \leqslant 4m$ and $\Delta(e) = 3m$; thus, $\rho(E) \leqslant \frac{4}{3}$.

Therefore, the only possibility is that case (iii) holds (with equality) in *all* cases. In other words, T' must have m Steiner points, each of degree 4, with all connecting lines of weight 1. However, this is only possible if these Steiner points are m of the s_j's, i.e., with $X_i = \bar{0}$ for all i, $Y_{j_k} = \bar{1}, Y_j = \bar{0}, j \neq j_k$. Consequently, the corresponding F_j's form an exact 3-cover of I_{3m}.

The preceding facts have as an immediate consequence the following result.

TABLE 1

Case[a]	Lower Bound on Length (E_k)	$\Delta(E_k)$	ρ
(i)	2	1	2
(ii)	3	2	$\frac{3}{2}$
(iii)	4	3	$\frac{4}{3}$
(iv)	5	3	$\frac{5}{3}$

[a]See Figure 1.

THEOREM 4.5

The Steiner problem for the N-cube Q_N is NP-*complete.*

COROLLARY

The Steiner problem in phylogeny is NP-complete.

5. SUMMARY

The problem of constructing phylogenies of maximum parsimony from nucleotide sequence data has been discussed and defined mathematically as the Steiner problem in phylogeny. It has been shown that this problem is NP-complete, which makes the existence of an effective algorithm for it a remote possibility. This means that it is likely that any SPP algorithm will require a number of elementary computational steps (and hence time) which is an exponential function of the size of the problem. Realistic biological studies typically involve over 20 taxa, each represented by a sequence of over 100 characters. Any construction procedure whose running time is not polynomially bounded will require an enormous amount of time. This result makes it much more realistic to attempt to devise heuristic solution procedures which run in reasonable computational time.

The authors are grateful to an anonymous reviewer who made a useful suggestion which was adopted in the discussion on intractability in Section 3.

REFERENCES

1 J. H. Camin and R. R. Sokal, *Evolution* 19:311–326 (1965).
2 J. Edmonds, *Canad. J. Math.* 17:449–467 (1965).
3 G. F. Estabrook, *J. Theoret. Biol.* 21:421–438 (1968).
4 W. M. Fitch and E. Margoliash, *Science* 155:279–284 (1967).
5 W. M. Fitch, *Syst. Zool.* 20:406–416 (1971).
6 W. M. Fitch and J. S. Farris, *J. Mol Evol.* 2:123–136 (1973).
7 W. M. Fitch, *Amer. Nat.* 111:223–257 (1977).
8 L. R. Foulds, M. D. Hendy, and David Penny, *J. Mol. Evol.* 13:127–149 (1979).
9 L. R. Foulds and R. W. Robinson, *Combinatorial Mathematics VII*, Lecture Notes in Mathematics, No. 829, Springer, Berlin, 1980, pp. 110–126.
10 L. R. Foulds and R. W. Robinson, *Combinatorial Mathematics VIII*, Lecture Notes in Mathematics, Springer, Berlin, 1981, pp 187–202.
11 M. R. Garey, R. L. Graham, and D. S. Johnson, *SIAM J. Appl. Math.* 32:835–859 (1977).
12 M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1979.
13 F. Harary, *Graph Theory*, 1st ed., Addison-Wesley, Reading, Mass., 1969.
14 J. A. Hartigan, *Biometrics* 29: 53–65 (1973).
15 M. D. Hendy, David Penny, and L. R. Foulds, *J. Theoret. Biol.* 71:441–452 (1978).
16 G. W. Moore, J. Barnabas, and M. Goodman, *J. Theoret. Biol.* 38:459–485 (1973).
17 David Penny, M. D. Hendy, and L. R. Foulds, *Biochem J.* 187:65–74 (1980).