Syst. Zool., 35(2):224-229, 1986

COMPUTATIONAL COMPLEXITY OF INFERRING PHYLOGENIES BY COMPATIBILITY

WILLIAM H. E. DAY[1] AND DAVID SANKOFF

Centre de Recherches Mathématiques, Université de Montréal, Case Postale 6128, Succursale A, Montréal, Québec H3C 3J7, Canada

Abstract.—A well-known approach to inferring phylogenies involves finding a phylogeny with the largest number of characters that are perfectly compatible with it. Variations of this problem depend on whether characters are: cladistic (rooted) or qualitative (unrooted); binary (two states) or unconstrained (more than one state). The computational cost of known algorithms that guarantee solutions to these problems increases at least exponentially with problem size; practical computational considerations restrict the use of such algorithms to analyzing problems of small size. We establish that the four basic variants of the compatibility problem are all *NP*-complete and, thus, are so difficult computationally that for them efficient optimal algorithms are not likely to exist. [Character compatibility; computational complexity; evolutionary tree; *NP*-complete; phylogenetic inference.]

Résumé.—Une approche bien connue à l'étude de l'évolution des espèces est basée sur la recherche de l'arbre phylogénétique avec lequel il y a un nombre maximal de caractères compatibles. Des variantes de ce problème impliquent des caractères soit cladistiques (avec racine), soit qualitatifs (sans racine); des caractères soit binaires (deux valeurs), soit sans contrainte (plus qu'une valeur). Les algorithmes connus pour résoudre ces problèmes exigent un temps d'ordinateur qui augmente de façon exponentielle en fonction de la taille du problème; ainsi, à toute fin pratique, on est contraint à des petits problèmes. Nous démontrons que les quatre variantes du problème de compatibilité sont toutes *NP*-complètes, donc il est presque certain qu'ils sont trop difficiles pour que des algorithmes efficaces puissent exister.

Le Quesne (1969:201) called a character uniquely derived if it "evolved only in one direction on a single occasion in its history," and he suggested that phylogenies be inferred by finding ones on which the largest possible numbers of characters can be uniquely derived. Constructing such a phylogeny is straightforward once an appropriate subcollection of characters has been identified, since a uniquely derived character determines a branch in the phylogeny. Thus, the interesting problems are to: determine what "appropriate" means in this context; and develop an efficient algorithm for finding a largest "appropriate" subcollection of characters.

Although the first problem (involving terminology) has been solved, the second (involving algorithm design) has not; below we provide evidence that its solution is unlikely. First, we summarize concepts germane to specifying appropriate subcollections of characters and define character compatibility for various character types. Next we discuss concepts germane to analyzing the computational complexity of decision problems; in particular, we define a class of computationally difficult decision problems that are called *NP*-complete. Finally, we establish, as our main results, that four basic character-compatibility decision problems are *NP*-complete.

CHARACTER COMPATIBILITY

Le Quesne's (1969, 1972) development of the uniquely-derived character concept has stimulated Estabrook, Johnson, and McMorris to investigate the mathematical foundations of character compatibility (Estabrook et al., 1975, 1976a, b; McMorris, 1975, 1977; Estabrook and McMorris, 1977, 1980). Their formalization of character compatibility has an elegant and provocative expression in terms of partial orders and finite sets (Estabrook and McMorris, 1980). A tree poset $(P, \leq)$ involves a set P

[1] Permanent address: Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland A1C 5S7, Canada.

partially ordered by a relation $\leq$ satisfying, for all elements a, b, and c of P, the tree condition that $a \leq c$ and $b \leq c$ must imply $a \leq b$ or $b \leq a$. A character-state tree $(P, \leq)$ is a tree poset in which every subset Q of P has a unique most-recent ancestral state p in P (i.e., $p \leq q$ for all q in Q and if any other $p' \leq q$ for all q in Q, then $p' \leq p$). The universal lower bound of a character-state tree is its ancestral state. Let S be a set of objects for which we wish to infer a phylogeny. A cladistic character on S is a map K from S to a character-state tree P such that for each p in P, there is a subset T of S for which p is the most-recent ancestral state of $\{K(t): t \text{ is in } T\}$; the requirement prevents P from containing unnecessary states. Informally, a character-state tree P describes the evolution of its character states, while a cladistic character K associates the objects in S with character states in P. A binary cladistic character is one whose character-state tree has just two states—ancestral and derived.

Character compatibility connotes a relationship between characters that exists if there is a phylogeny on which they could evolve with each state arising no more than once. Character compatibility can be defined using set-theoretic concepts. A tree of subsets of S is a set $\mathbf{K}$ of nonempty subsets of S such that: S is in $\mathbf{K}$; if A and B are in $\mathbf{K}$, then their set intersection is in $\{\varnothing, A, B\}$. The set of possible cladistic characters on S is in one-to-one correspondence with the set of possible trees of subsets of S (Estabrook and McMorris, 1980:Theorem 1). Let K and L be cladistic characters with corresponding trees of subsets $\mathbf{K}$ and $\mathbf{L}$; then K is a refinement of L if $\mathbf{L}$ is a subset of $\mathbf{K}$. A collection of cladistic characters is compatible if a cladistic character exists that is a refinement of every character in the collection. A major result, which Felsenstein (1982) called the Pairwise Compatibility Theorem, is that a collection of cladistic characters is compatible if and only if the characters are compatible by pairs (Estabrook et al., 1976b:Theorem 2.4; Estabrook and McMorris, 1980:Theorem 4).

The concept of cladistic character can be generalized by dropping the requirements that its character-state tree be rooted or that it have a character-state tree at all. In the latter case, a qualitative character is defined to be a map K from S to a set P of character states; since no other restrictions are imposed, a qualitative character has states, but not a character-state tree. For a binary qualitative character, P is a two-element set. A collection of qualitative characters is compatible if their character-state sets can be ordered to make the collection a compatible set of cladistic characters. The Pairwise Compatibility Theorem does not hold for qualitative characters (Fitch, 1975; McMorris, 1975), but it does hold for binary qualitative characters (McMorris, 1977:Theorem 1).

DECISION PROBLEMS

Consider the problem of finding a clique in a graph (Karp, 1972; Garey and Johnson, 1979:194).

Clique

Instance.—A graph $G = (V, E)$ consisting of a set V of m vertices and a set E of edges joining some of these vertices; and a positive integer $B \leq m$.

Question.—Does G contain a clique of size B or more (i.e., a subset V' of V containing B or more vertices such that every two vertices in V' are joined by an edge in E)?

Clique is a decision problem since each solution is either "yes" or "no." Its description illustrates a standard format for specifying decision problems: one part specifies a generic instance of the problem; the other states a yes-no question in terms of the generic instance.

Although finding solutions to Clique seems difficult, verifying solutions to Clique is easily done in polynomial time (i.e., by an algorithm requiring a number of steps that can be bounded in advance by a polynomial function of m). *NP* is the set of all decision problems having polynomial time verification algorithms, and so Clique is in *NP*. Clique is among the most difficult *NP* problems to solve because every other *NP* problem is a special case of Clique (i.e., for any such problem there is a polynomial-time procedure to convert each instance of that problem into an equivalent instance of Clique). Thus,

any polynomial-time algorithm solving Clique would solve every *NP* problem in polynomial time. Such difficult *NP* problems are called *NP*-complete.

NP-completeness is defined in the following way. Let D_i denote the set of all instances of a decision problem P_i. A polynomial transformation from P_i to P_k is a map f from D_i to D_k such that: f is computable by a polynomial-time algorithm; for each instance l in D_i, the P_i solution for l is "yes" if and only if the P_k solution for $f(l)$ is "yes." In such cases, we say that P_i transforms to P_k; informally, P_i is a special case of P_k. If P_i transforms to P_k and also P_k transforms to P_i, we say that P_i and P_k are computationally equivalent. A decision problem P is *NP*-complete if P is in *NP* and P' transforms to P for all P' in *NP*. Clique is *NP*-complete (Karp, 1972). The *NP*-completeness of any other decision problem P can be established by showing that P is in *NP* and that Clique, or any other *NP*-complete problem, transforms to P (Garey and Johnson, 1979:38). This proof strategy has been used to establish that several problems in phylogenetic systematics are *NP*-complete. Graham and Foulds (1982) showed the Steiner Problem in Phylogeny (SPP) to be *NP*-complete by transforming Exact Cover By 3-Sets, a known *NP*-complete problem (Garey and Johnson, 1979: 221), to it. Day (1983) showed the Wagner Tree Problem and the Additive Evolutionary Tree Problem to be *NP*-complete by transforming SPP to each of them.

Problems of inferring phylogenies by character compatibility can be stated as decision problems. Four interesting cases vary according to whether characters in the problem are: binary or unconstrained; and cladistic or qualitative. Although we defined characters in terms of partial orders, they are specified conveniently by the usual character-by-object matrix. Thus, the four basic compatibility decision problems have these formulations.

Binary Cladistic Compatibility (BCC)

Instance.—Number n of objects; description of m binary cladistic characters by an m-by-n character-by-object matrix X; positive integer $B \leq m$.

Question.—Does the collection of characters described by X have a compatible subcollection of B or more characters?

Unconstrained Cladistic Compatibility (UCC)

Instance.—Number n of objects; description of m cladistic characters by an m-by-n character-by-object matrix X; positive integer $B \leq m$.

Question.—Same as for BCC.

Binary Qualitative Compatibility (BQC)

Instance.—Number n of objects; description of m binary qualitative characters by an m-by-n character-by-object matrix X; positive integer $B \leq m$.

Question.—Same as for BCC.

Unconstrained Qualitative Compatibility (UQC)

Instance.—Number n of objects; description of m qualitative characters by an m-by-n character-by-object matrix X; positive integer $B \leq m$.

Question.—Same as for BCC.

NP-COMPLETENESS RESULTS

NP-completeness proofs often focus on special cases of the problem of interest. We focus on binary compatibility problems, since they transform to corresponding unconstrained problems.

Proposition 1.—BCC transforms to UCC; BQC transforms to UQC.

Proof.—Use an identity mapping f to associate each instance l of the first problem with an instance $f(l)$ of the second problem. This mapping preserves "yes" solutions since every binary character also satisfies the definition of an unconstrained character. ■

Our first nontrivial result establishes the computational equivalence of binary compatibility problems.

Proposition 2.—BQC and BCC are computationally equivalent.

Proof.—BQC transforms to BCC by the following argument. To each instance $l = (m, n, X, B)$ of BQC there corresponds an instance $f(l) = (m, n, X', B)$ of BCC. X' is constructed from X by requiring the most frequently occurring state of each character in X to become the ancestral state of that character in X'. To show that f preserves "yes" solutions, let Y be a subcollection of characters in l, with Y' the cor-

responding subcollection of characters in $f(l)$. Then Y is a compatible collection of binary qualitative characters: if and only if the characters in Y are pairwise compatible (McMorris, 1977:Theorem 1); if and only if the characters in Y' are pairwise compatible (McMorris, 1977:Lemma); if and only if Y' is a compatible collection of cladistic characters (Estabrook et al., 1976b: Theorem 2.4).

BCC transforms to BQC by the following argument. To each instance $l = (m, n, X, B)$ of BCC there corresponds an instance $f(l) = (m, 2n, X', B)$ of BQC. X' is constructed by appending to X the columns for n new objects exhibiting the ancestral states of the characters in X. Let Y and Y' be defined as before. Y is a compatible set of cladistic characters if and only if Y' is a compatible set of cladistic characters; that f preserves "yes" solutions now follows using McMorris' (1977) Lemma and Theorem 1. ■

We use the Pairwise Compatability Theorem to convert the problem of finding a maximal set of compatible cladistic characters into the problem of finding a clique in a graph.

Proposition 3.—BCC and UCC each transform to Clique.

Proof.—To each instance $l = (m, n, X, B)$ of BCC or UCC there corresponds an instance $f(l) = (G, B)$ of Clique. G is a graph with vertices $V = \{v_1, \ldots, v_m\}$ corresponding to the cladistic characters $K_1, \ldots, K_m$ described by X, and with edges $E = \{\{v_i, v_k\} : v_i \text{ and } v_k \text{ are in } V, K_i \text{ and } K_k \text{ are compatible}\}$. The Pairwise Compatibility Theorem (Estabrook et al., 1976b:Theorem 2.4) ensures that G contains a clique of B or more vertices if and only if X contains a compatible subcollection of B or more cladistic characters. Thus the mapping f preserves "yes" solutions. ■

As a consequence of Proposition 3, algorithms finding maximal cliques in graphs can be employed to find collections of cladistic characters or (as a consequence of Proposition 2) binary qualitative characters. Viewed computationally, this approach is ineffective since no efficient algorithm is known to find maximal cliques in graphs: recall that Clique is *NP*-complete and so is a hardest problem in *NP*. But Proposition 3 does establish that BCC and UCC are special cases of Clique; consequently BCC and UCC problems may have special structure enabling them to be solved by polynomial time algorithms, even though such algorithms are not known for the perhaps more general Clique problem. The next results dash this hope by establishing that Clique, BCC, and UCC are all computationally equivalent.

Proposition 4.—Clique transforms to BCC.

Proof.—To each instance $l = (G, B)$ of Clique, where $G = (V, E)$, there corresponds an instance $f(l) = (m, n, X, B)$ of BCC. Set $m = |V|$; if $V = \{v_1, \ldots, v_m\}$, then X contains rows to describe characters $K_1, \ldots, K_m$. Set $n = 3m(m - 1)/2$; X has three columns for each unordered pair of vertices in V. X is specified so that K_i and K_k are compatible if and only if $\{v_i, v_k\}$ is in E. Initialize all entries in X to 0. We use the result (Estabrook et al., 1976b:Theorem 2.3) that binary cladistic characters K_i and K_k are incompatible if and only if all three of the elements (0, 1), (1, 1), (1, 0) are in $(K_i \times K_k)(S)$. For each edge $\{v_i, v_k\}$ not in E, enter in the three columns for that edge: 011 in the row for K_i; 110 in the row for K_k. Then for V' a subset of V and F' the corresponding subcollection of the characters described by X, V' is a clique in G: if and only if E has an edge between every pair of vertices in V' (by definition); if and only if every pair of characters in F' is compatible (by construction); if and only if F' is a compatible collection of cladistic characters (Estabrook et al., 1976b:Theorem 2.4). Thus, the mapping f preserves "yes" solutions. ■

Proposition 5.—BCC, BQC, UCC, and UQC are *NP*-complete.

Proof.—Clearly the four problems are in *NP*. Clique is *NP*-complete (Karp, 1972). Propositions 1, 2, and 4 establish that Clique transforms to each of the four problems. ■

DISCUSSION

These results are discouraging, computationally, because they support the con-

jecture that efficient algorithms cannot be designed to obtain globally optimal solutions for important compatibility problems. At the same time, they may encourage practitioners to explore other approaches which require only reasonable amounts of computing time. For example, algorithms producing locally optimal solutions can be very efficient, and the results can be used with some confidence if an algorithm converges to the same local solution from a variety of starting points. As Graham and Foulds (1982:137) observed:

> . . . 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 [efficient].

Alternatively, some algorithms producing global optima, although inefficient because they require exponential execution time in the worst case, may in fact be relatively efficient for most sets of data, including those encountered in practice. For example, algorithms finding all the maximal cliques in a graph can be shown to require exponential execution time in the worst case, but their average execution time, under reasonable probabilistic hypotheses about the graphs to be processed, may be modest, or the probability that they terminate quickly may be quite high. When used for compatibility investigations, algorithms finding all maximal cliques seem to exhibit such behavior.

Still another approach to circumventing excessive computational requirements is to consider somewhat constrained problems. There are many ways to constrain the clique and compatibility problems described in this paper. For example, if either m, the number of characters, or n, the number of objects, is constrained to be less than some fixed constant, then it is easy to show that the problems become globally solvable in time which is polynomial in n or m, respectively. Much weaker constraints can also achieve this type of efficiency.

We conclude by mentioning several open problems concerning the inference of phylogenies. Now that the *NP*-completeness of the Wagner Tree and related problems has been established (Graham and Foulds, 1982; Day, 1983), as well as the compatibility problems considered herein, it remains to prove (or disprove) similar results for the Camin-Sokal (Camin and Sokal, 1965), Dollo (Le Quesne, 1974, 1977; Farris, 1977a, b), and chromosome-inversion (Farris, 1978) parsimony methods for inferring phylogenies.

ACKNOWLEDGMENTS

We thank J. Felsenstein and G. D. Schnell for their criticisms of a draft of this manuscript. The Natural Sciences and Engineering Research Council of Canada partially supported this research through individual operating grants to W. H. E. Day (A4142) and D. Sankoff (A8867), as well as through an infrastructure grant to D. Sankoff, R. J. Cedergren, and G. Lapalme (A3092).

REFERENCES

CAMIN, J. H., AND R. R. SOKAL. 1965. A method for deducing branching sequences in phylogeny. Evolution, 19:311–326.

DAY, W. H. E. 1983. Computationally difficult parsimony problems in phylogenetic systematics. J. Theor. Biol., 103:429–438.

ESTABROOK, G. F., C. S. JOHNSON, JR., AND F. R. MCMORRIS. 1975. An idealized concept of the true cladistic character. Math. Biosci., 23:263–272.

ESTABROOK, G. F., C. S. JOHNSON, JR., AND F. R. MCMORRIS. 1976a. A mathematical foundation for the analysis of cladistic character compatibility. Math. Biosci., 29:181–187.

ESTABROOK, G. F., C. S. JOHNSON, JR., AND F. R. MCMORRIS. 1976b. An algebraic analysis of cladistic characters. Discrete Math., 16:141–147.

ESTABROOK, G. F., AND F. R. MCMORRIS. 1977. When are two qualitative taxonomic characters compatible? J. Math. Biol., 4:195–200.

ESTABROOK, G. F., AND F. R. MCMORRIS. 1980. When is one estimate of evolutionary relationships a refinement of another? J. Math. Biol., 10:367–373.

FARRIS, J. S. 1977a. Phylogenetic analysis under Dollo's law. Syst. Zool., 26:77–88.

FARRIS, J. S. 1977b. Some further comments on Le Quesne's methods. Syst. Zool., 26:220–223.

FARRIS, J. S. 1978. Inferring phylogenetic trees from chromosome inversion data. Syst. Zool., 27:275–284.

FELSENSTEIN, J. 1982. Numerical methods for inferring evolutionary trees. Q. Rev. Biol., 57:379–404.

FITCH, W. M. 1975. Toward finding the tree of maximum parsimony. Pages 189–230 *in* The eighth international conference on numerical taxonomy (G. F. Estabrook, ed.). W. H. Freeman, San Francisco.

GAREY, M. R., AND D. S. JOHNSON. 1979. Computers and intractability. W. H. Freeman, San Francisco.

GRAHAM, R. L., AND L. R. FOULDS. 1982. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. Math. Biosci., 60:133–142.

KARP, R. M. 1972. Reducibility among combinatorial problems. Pages 85–103 *in* Complexity of computer computations (R. E. Miller and J. W. Thatcher, eds.). Plenum, New York.

LE QUESNE, W. J. 1969. A method of selection of characters in numerical taxonomy. Syst. Zool., 18: 201–205.

LE QUESNE, W. J. 1972. Further studies based on the uniquely derived character concept. Syst. Zool., 21: 281–288.

LE QUESNE, W. J. 1974. The uniquely evolved character concept and its cladistic application. Syst. Zool., 23:513–517.

LE QUESNE, W. J. 1977. The uniquely evolved character concept. Syst. Zool., 26:218–220.

MCMORRIS, F. R. 1975. Compatibility criteria for cladistic and qualitative taxonomic characters. Pages 399–415 *in* The eighth international conference on numerical taxonomy (G. F. Estabrook, ed.). W. H. Freeman, San Francisco.

MCMORRIS, F. R. 1977. On the compatibility of binary qualitative taxonomic characters. Bull. Math. Biol., 39:133–138.

Received 10 December 1985; accepted 27 May 1986.