COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 9, No. 4, pp. 295-302, 2009

001

THE THREE-STATE PERFECT PHYLOGENY PROBLEM REDUCES TO 2-SAT*

DAN GUSFIELD† AND YUFENG WU‡

Abstract. We extend a structural result by A. Dress and M. Steel [3], to show that the three-state Perfect Phylogeny problem reduces in polynomial time to the classic 2-SAT problem. We also give a more expanded exposition of the proof of the structural result from [3]. We hope this note will encourage additional researchers to try to solve the central open question: finding simple efficient solutions to the k-state Perfect Phylogeny problem for $k > 3$.

1. Introduction. In the k-state Perfect Phylogeny problem, we are given an n by m matrix M with integers from the set $K = \{1, 2, ..., k\}$. The problem arises in phylogenetics, so we refer to each row of M as a *taxon* (plural *taxa*), to each column of M as a *character*, and to each value in a column c as a *state* of character c. A Perfect Phylogeny for M is an undirected tree T with n leaves, where each leaf is labeled by a distinct taxon of M, and each internal node of T is labeled by a vector in K^m (which need not be in M), such that for every character c and every state i the nodes labeled with state i for character c form a *connected* subtree, $T_c(i)$, of T. Clearly then, for any character c and states $i \neq j$, the subtrees $T_c(i)$ and $T_c(j)$ of T are node disjoint. An example is shown in Figure 1. The k-state Perfect Phylogeny Problem is to determine, for input M, if there is a k-state perfect phylogeny for M.

If none of the parameters k, n or m is fixed (so k can grow with n), then the k-state perfect phylogeny problem is NP-complete [2, 11]. In contrast, if k is any fixed integer, independent of n, then the problem can be solved in time that is polynomial in n and m. In fact, for $k = 2$, the problem can be solved in linear time [5]. A polynomial-time solution for $k = 3$ was shown in [3]; a polynomial-time solution for $k = 3$ or 4 was shown in [7]; and a polynomial bound for any *fixed* k was shown in [1]. The later result was improved in [8] to a time bound of $O(2^{2k}nm^2)$. An excellent survey of most of these results appears in [4].

The polynomial-time algorithm for $k = 3$, developed in the paper by A. Dress and M. Steel [3] is very simple in comparison to the methods for $k > 3$ [7, 1, 8][1]. The work in this note comes out of an effort to find equally simple methods for $k > 3$, or

*Dedicated to Michael Waterman on the occasion of his 67th birthday.

†Department of Computer Science, University of California, Davis. E-mail: gusfield@cs.ucdavis.edu

‡Department of Computer Science and Engineering, University of Connecticut. E-mail: ywu@engr.uconn.edu

[1]We do not know of any working implementations of these latter methods. We have implemented a slower worst-case, yet polynomial-time, version of the method in [8]; it contains more than one thousand lines of C.

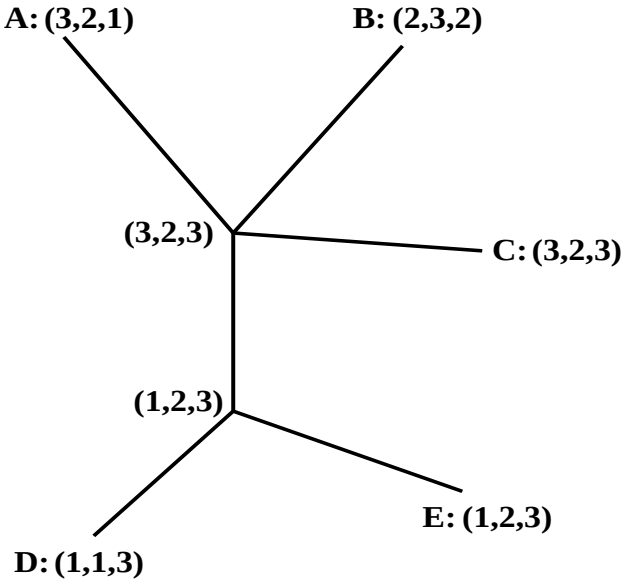

FIG. 1. *A three-state perfect phylogeny with $n = 5, m = 3$. The input M consists of the five vectors that label the leaves of the tree. The subtree $T_3(3)$ contains the leaves labeled C,D,E, and the two interior nodes.*

to find simple ways to explain that the k-state Perfect Phylogeny problem with any fixed k can be solved in polynomial time.

In this note we show that a result developed by A. Dress and M. Steel [3] can be extended to give a simple, efficient reduction of any instance of the 3-state perfect phylogeny problem to an instance of the classic 2-SAT problem[2] which is well-known to be solvable in polynomial time. This reduction is a small extension of the main result in [3], but it is valuable to explicitly state this observation for three reasons: a) to allow the practical application of the extensive literature on the 2-SAT problem to the 3-state perfect phylogeny problem; b) to add to the set of problems that have a simple, efficient reduction to 2-SAT; and c) to emphasize the distinction of the 3-state perfect phylogeny problem from the k-state problem for $k > 3$, where we do not know any natural reduction to a classic problem in P. More starkly, for any fixed k, the k-state Perfect Phylogeny problem *can* be reduced in polynomial time to the SAT problem restricted to a subclass of boolean expressions for which the SAT decision problem is in P (similar to the case of 2-SAT). This follows because the k-state Perfect Phylogeny problem is in P for any fixed k, and the problem HORN-SAT is complete for P, and is solved by a very simple polynomial-time algorithm. HORN-SAT is the

[2]The Dress-Steel paper actually shows a more general result, that of determining for any k, whether there is a k-state perfect phylogeny T for M, where each subtree $T_c(i)$ is a *star*. That problem can also be efficiently reduced to 2-SAT, but we leave that to the reader.

satisfiability problem where each clause in the boolean expression can have at most one positive literal. However, even for $k = 4$ we do not know a simple, natural way to reduce the k-state Perfect Phylogeny problem to the HORN-SAT problem.

A final, pedagogical contribution of this note is to provide a more expanded exposition, but following the same logical lines, of the central argument in [3]. A better understanding of this result may encourage additional researchers to try to solve the central open question: finding simple efficient solutions to the Perfect Phylogeny problem for $k > 3$. A wider understanding of multi-state Perfect Phylogeny is also important because of the increasing availability of multi-state molecular marker data, and particularly the expanding role of multi-state data in population genomics.

For other recent results on three-state perfect phylogeny, see [9], and for recent results on more than three states, see [6].

2. The Dress-Steel solution to the 3-state Perfect Phylogeny Problem. For this exposition, create another matrix $\overline{M}$ derived from M, with three characters $c(1), c(2), c(3)$, for each character c in M. Note that c is a *variable*. All the taxa that have state i for c in M are given state 1 for character $c(i)$ in $\overline{M}$, and the other taxa are given state 0 for $c(i)$. So, the original input matrix M is recoded as a binary matrix $\overline{M}$ with three expanded characters for each character in M. Two columns in $\overline{M}$ (and their corresponding characters) are called "incompatible" if and only if the two columns contain all four binary pairs 0,0; 0,1; 1,0; 1,1, and they are otherwise called "compatible". The main structural result in [3], interpreted in terms of $\overline{M}$ is:

THEOREM 2.1. *Given matrix M with $k = 3$, there is a perfect phylogeny for M, if and only if there is a set of characters S of $\overline{M}$ which are pairwise compatible, and where for each character c in M, S contains at least two of the characters $c(1), c(2), c(3)$.*

In [3], a polynomial-time algorithm is given to find an appropriate set S, if one exits. That algorithm can be seen to be isomorphic to the classic (resolution) algorithm that determines if a 2-SAT expression is satisfiable, and therefore suggests that there might be an explicit reduction of the 3-state Perfect Phylogeny problem to 2-SAT. We next give such an explicit reduction.

Reduction to 2-SAT. Given M, we form the following 2-SAT expression $E(M)$ containing one variable $x_c(i)$ for each character $c(i)$ in $\overline{M}$. For each character c in M, create the subformula $F_c = [(x_c(1) \vee x_c(2))] \wedge [(x_c(1) \vee x_c(3)] \wedge [(x_c(2) \vee x_c(3))]$, and create the subformula $F_1(M)$ consisting of the conjunction of all of the F_c subformulas. $F_1(M)$ contains only two literals per clause, and is satisfied if and only if at least two of the variables $x_c(1), x_c(2), x_c(3)$ are set to true, for each character c.

Next, for every *incompatible* pair of characters $c(i), c'(j)$ in $\overline{M}$, where c and c' are different characters in M and i and j are each in $\{1, 2, 3\}$, create the clause $[\neg x_c(i) \vee \neg x_{c'}(j)]$, and create the sub formula $F_2(M)$ consisting of the conjunction of all these clauses. Subformula $F_2(M)$ contains only two literals per clause, and is satisfied

if and only if no pair of variables that correspond to a pair of incompatible characters in $\overline{M}$, are both set true. Finally, create the 2-SAT expression $E(M) = F_1(M) \wedge F_2(M)$.

Clearly, if there is a set of characters S satisfying the requirements in Theorem 2.1, then by giving each variable corresponding to a character in S a value of "true", expression $E(M)$ is satisfied. Conversely, if $E(M)$ is satisfied, then by selecting S to be the set of characters of $\overline{M}$ whose corresponding variables are set true, then the set S satisfies the requirements of Theorem 2.1.

Note that if $E(M)$ is satisfied by an assignment that sets all three of the variables $x_c(1), x_c(2), x_c(3)$ true, for some character c in M, then any one of those three variables can be reset to false while still satisfying $E(M)$. Therefore, we can sharpen Theorem 2.1 as follows:

THEOREM 2.2. *Given matrix M with $k = 3$, there is a perfect phylogeny for M, if and only if there is a set of characters S of $\overline{M}$ which are pairwise compatible, and where for each character c in M, S contains* exactly *two of the characters $c(1), c(2), c(3)$.*

2.1. An expanded exposition of the proof of Theorem 2.1. For pedagogical purposes, in this section we give an expanded exposition of the proof of Theorem 2.1 in [3]. The exposition is expanded, but the proof follows the same logical lines as in [3].

Facts about splits. A bipartition $(X, \overline{X})$ of a set $\mathcal{Z}$ is called a "split". Two splits $(X, \overline{X}), (Y, \overline{Y})$ are called "incompatible" if and only if all of the sets $X \cap Y, \overline{X} \cap Y, X \cap \overline{Y}, \overline{X} \cap \overline{Y}$ are nonempty, and are otherwise called "compatible". When $\mathcal{Z}$ is the set of taxa in M, these definitions are consistent with the definitions given earlier for compatible and incompatible characters in $\overline{M}$. Also, the removal of an edge e from a perfect phylogeny T splits T into two connected components, creating a bipartition of the leaves of T, and hence defines a split of the taxa, called the "split defined by e". For any character c in M, we use $X_c(i)$ to denote the set of taxa which have state i for character c. Hence, $X_c(i)$ is also the set of taxa which has state 1 in the column associated with $c(i)$ in $\overline{M}$.

The most fundamental fact about leaf-labeled trees is the *Splits Equivalent Theorem* [10]: Given a set of splits $\mathcal{F}$, defined on a set $\mathcal{Z}$ of size n, there is an undirected tree T with n leaves where each leaf is labeled with a distinct element of $\mathcal{Z}$, and where T contains edges that define the set of splits $\mathcal{F}$ (possibly with other edges), if and only if every pair of splits in $\mathcal{F}$ is compatible. As an immediate corollary, if e and e' are two edges in a tree T, then the two splits defined by e and e' must be compatible. We can now prove Theorem 2.1.

Proof of Theorem 2.1. Suppose there is a 3-state perfect-phylogeny T for M. For any character c of M, the subtrees $T_c(1), T_c(2)$ and $T_c(3)$ are node disjoint and contain all the nodes of T. Now for each character c, contract, in T, all of the nodes of $T_c(i)$ to a single node. The resulting graph must be a path P_c with three nodes; we label

each node v in P_c with the distinct state (1, 2, or 3) of the nodes that contract to v. For example, in the perfect-phylogeny T shown in Figure 1, if we contract each of the subtrees $T_3(1), T_3(2), T_3(3)$ to a single node, we get a path P_3 labeled with end nodes 1 and 2 and with interior node labeled 3.

In general, we use i and j to denote the state-labels of the two nodes at the leaves of P_c. Since P_c is a path with two edges, there is an edge e in P_c the node labeled i from the interior node and the node labeled j. Edge e is an uncontracted edge from T, and so edge e separates all the taxa with state i for character c from all the taxa with the other two states, and hence defines a bipartition of the taxa and the split $(X_c(i), \overline{X_c(i)})$. Similarly, there is also an edge in T that defines the split $(X_c(j), \overline{X_c(j)})$. Then, for character c, select characters $c(i)$ and $c(j)$ to be in S. Repeating this for each character c of M selects a set S of characters of $\overline{M}$ that contains exactly two expanded characters for each character c in M. Further, since each selected split is defined by an edge in T, and every pair of splits defined by edges in T are compatible, the characters in S are pairwise compatible and the necessary direction of Theorem 2.1 is proved.

Conversely, suppose there is a set of characters S in $\overline{M}$ satisfying the conditions of Theorem 2.1. Let $\mathcal{Z}$ denote the set of taxa in M. By construction, each character in $\overline{M}$ defines a split of the taxa $\mathcal{Z}$ in M, and so S defines a set of pairwise-compatible splits of the taxa. For a taxon s in M, the "trivial split" for s is the bipartition $\{s, \mathcal{Z} - s\}$, which is clearly compatible with any other split. We augment the splits defined by S with these n trivial splits, and call the resulting set of splits S'. By the Splits-Equivalent Theorem there is some tree T' with n leaves, each labeled with a distinct taxon in $\mathcal{Z}$, and containing edges that define the splits in S'. We can assume that each edge in T' actually defines one of the splits in S', by contracting any edge that does not define such a split. Also, we can assume that no internal node of T' has degree two, since otherwise two neighboring edges define the same split, in which case one edge can be contracted. We now show how to map the taxa to leaves of T and how to label the interior nodes in T' so that T becomes a perfect phylogeny for M.

Because of the trivial splits in S', each taxon in $\mathcal{Z}$ labels a leaf of T', satisfying one requirement for a perfect-phylogeny for M. We next need to show how to label the interior nodes of T' so that for every character c and every state i for c, $T'_c(i)$ is a connected subtree of T'. For a character c in M, suppose, without loss of generality, that characters $c(1)$ and $c(2)$ are in S', and let $e(1)$ and $e(2)$ be the edges in T' that define the splits $(X_c(1), \overline{X_c(1)})$, and $(X_c(2), \overline{X_c(2)})$. Removal of $e(1)$ from T' creates two connected subtrees, one which contains all and only the taxa in $X_c(1)$ labeling its leaves. Label each of the nodes in that subtree with state 1 for character c, defining subtree $T'_c(1)$. Define T'' as the tree T' after the removal of all nodes and edges in $T'_c(1)$. Clearly, T'' contains all the leaves labeled by taxa in $X_c(2)$. T'' also

contains edge $e(2)$; otherwise $e(2)$ would be an edge in $T'_c(1)$ and since all interior nodes have degree three or more, there would be a leaf labeled 1 on both sides of $e(2)$, contradicting the assumption that $e(2)$ defines the split $(X_c(2), \overline{X_c(2)})$. So, removal of $e(2)$ from T'' defines two connected subtrees of T', one which contains all and only the taxa in $X_c(2)$; label the nodes of that subtree with state 2 for character c, defining $T'_c(2)$. Removing $T'_c(2)$ from T'' leaves a connected subtree of T' that must contain all and only the leaves labeled by taxa in $X_c(3)$. Label the nodes in that subtree with state 3 for character c, creating $T'_c(3)$. These three subtrees are node disjoint and show that character c obeys the convexity requirement. Since the argument holds for any c, we conclude that T' (with interior nodes labeled as above) is a 3-state perfect phylogeny for M.

3. Open Problem. The main open problem is to find a simple, polynomial-time reduction of the k-state Perfect Phylogeny problem, for any fixed k (or even for $k = 4$) to a classic problem in P. That would provide, in contrast to the algorithms in [1, 8], a simple demonstration that the k-state Perfect Phylogeny problem, for fixed k, is in P. This goal seems plausible since, as stated in the introduction, for any fixed k, the k-state Perfect Phylogeny problem can be reduced in polynomial-time to the HORN-SAT problem (which is in P and can be solved with a very simple algorithm), but the reduction is not "simple" or "natural". There is also a simple, polynomial-time reduction of the k-state Perfect Phylogeny problem for any fixed k, to the SAT problem where the boolean expressions only contain Horn-clauses and clauses with at most two literals. However, that reduction does not seem helpful because any instance of SAT can also be reduced in polynomial time to such a form of SAT, and hence no polynomial-time algorithm is expected that can solve all SAT problems of this form.

4. Acknowledgement. We thank Mike Steel for helpful discussion and comments. This research was partially supported by NSF grants SEI-BIO 0513910, CCF-0515378, IIS-0803564 and IIS-0803440.

REFERENCES

[1] R. Agarwala and D. Fernandez-Baca. *A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed.* SIAM J. on Computing, 23(1994), pp. 1216–1224.

[2] H. Bodlaender, M. Fellows, and T. Warnow. *Two strikes against perfect phylogeny.* Proc. of the 19'th Inter. colloquium on Automata, Languages and Programming, pages 273–283, 1992.

[3] A. Dress and M. Steel. *Convex tree realizations of partitions.* Applied Math Letters, 5(1993), pp. 3–6.

[4] D. Fernandez-Baca. *The perfect phylogeny problem.* In: D.Z. Du and X. Cheng, editors, Steiner Trees in Industries. Kluwer Academic Publishers, 2000.

[5] D. Gusfield. *Efficient algorithms for inferring evolutionary history.* Networks, 21(1991), pp. 19–28.

[6] D. Gusfield. *The multi-state perfect phylogeny problem: Solutions via integer linear programming and chordal graph theory.* In: Proc. of RECOMB 2005: The 13th Ann. International Conference Research in Computational Molecular Biology. Springer, LNBI, 2009.

[7] S. Kannan and T. Warnow. *Inferring evolutionary history from DNA sequences.* SIAM J. on Computing, 23(1994), pp. 713–737.

[8] S. Kannan and T. Warnow. *A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed.* SIAM J. on Computing, 26(1997), pp. 1749–1763.

[9] F. Lam, D. Gusfield, and S. Sridhar. *Generalizing the four gamete condition and splits equivalence theorem: Perfect phylogeny on three state characters.* In: Proc. of WABI 2009, Lecture Notes in Computer Science. Springer, 2009.

[10] C. Semple and M. Steel. *Phylogenetics.* Oxford University Press, UK, 2003.

[11] M. Steel. *The complexity of reconstructing trees from qualitative characters and subtrees.* J. of Classification, 9(1992), pp. 91–116.