Two Strikes Against Perfect Phylogeny

Hans L. Bodlaender* Mike R. Fellows† Tandy J. Warnow‡

Abstract

One of the major efforts in molecular biology is the computation of phylogenies for species sets. A longstanding open problem in this area is called the Perfect Phylogeny problem. For almost two decades the complexity of this problem remained open, with progress limited to polynomial time algorithms for a few special cases, and many relaxations of the problem shown to be NP-Complete. From an applications point of view, the problem is of interest both in its general form, where the number of characters may vary, and in its fixed-parameter form. The Perfect Phylogeny problem has been shown to be equivalent to the problem of triangulating colored graphs[30]. It has also been shown recently that for a given fixed number of characters the yes-instances have bounded treewidth[45], opening the possibility of applying methodologies for bounded treewidth to the fixed-parameter form of the problem. We show that the Perfect Phylogeny problem is difficult in two different ways. We show that the general problem is NP-Complete, and we show that the various finite-state approaches for bounded treewidth cannot be applied to the fixed-parameter forms of the problem.

1 Introduction

Historically, one of the major efforts in molecular biology has been the computation of phylogenetic trees, or *phylogenies*, which describe the evolution of a set of species from a common ancestor. A *phylogeny* for the set S of species, is a rooted tree in which the leaves represent the species in S and the internal nodes of the tree represent the ancestral species. The computational complexity of determining a most-likely phylogeny for the species set then depends, among other things, on how the species set is described. One of the standard models uses *characters* to describe species. Here, a character is an equivalence relation on the species set, partitioning the set into the different *character states*. Under this model, a proposed phylogeny will also assign character states to each of the hypothesized species indicated by the internal nodes. The desired property for the phylogeny is the following:

> For each state of each character, the set of nodes in the tree having that state should form a connected component.

When the phylogeny has this property, it is said to be *perfect*, and the characters are also said to be *perfectly compatible*. The Perfect Phylogeny problem[28] (in short: PP; also known as the Character Compatibility problem[21]) is then as follows.

> **Perfect Phylogeny:** For a given set of characters defining a species set S, does a perfect phylogeny exist?

*Department of Computer Science, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. This author was partially supported by the Esprit II Basic Research Actions Program of the EC under contract no. 3075 (project ALCOM).
†Computer Science Department, University of Victoria, Victoria, B.C., Canada.

‡Department of Mathematics, University of Southern California, Los Angeles, California, USA. This author was partially supported by NSF Grants no. CCR88-13632 and no. DMS90-05833

If the number of characters is a fixed constant k, we call the problem the k-Perfect Phylogeny problem.

This approach to constructing phylogenies was probably first discussed in the biological literature in the 1960's (see [13, 58] for two of the earliest papers, and the series of papers by LeQuesne [38, 39, 40, 41]), but was given its precise mathematical formulation by Estabrook and others in a series of papers beginning in 1972 (see [16, 17, 18, 19]). In 1974, Buneman showed[12] that the Perfect Phylogeny problem reduced to a graph-theoretic problem, which we call the *Triangulating Colored Graphs* problem (or TCG). A graph is said to be *triangulated* if every induced cycle contains at least four vertices. The Triangulating Colored Graphs problem is:

Input: Graph $G = (V, E)$, coloring $c : V \to Z$.
Question: Does there exist a supergraph $G' = (V, E')$ of G which is properly colored by c and which is triangulated?

If I is the instance of the Perfect Phylogeny problem, and G_I the corresponding instance of the Triangulating Colored Graphs problem, then vertices of G_I correspond to the character states of I, with states of the same character having the same color. Two vertices are adjacent if their corresponding character states share a species in common. Thus, the number of colors of TCG corresponds to the number of characters in the Perfect Phylogeny problem.

In 1990, Kannan and Warnow [30] showed that these two problems were polynomially equivalent. Linear time algorithms for the case of two and three-colored graphs have been found [7, 30] (corresponding to two and three character compatibility), and a polynomial time algorithm for the case of quaternary characters has been found [31]). The latter algorithm can be used to construct phylogenetic trees from DNA sequences. For the general case, the best that is known is an $O(n^{k+1})$ algorithm to triangulate (if possible) a k-colored graph [45].

In this paper we will show the following:

Theorem A: Perfect Phylogeny is NP-complete.

Theorem B: k–PERFECT PHYLOGENY is not finite-state for bounded treewidth, for $k \geq 4$.

The significance of Theorem B is the following. There are a large number of papers, that show that many problems, that are often combinatorially hard, become linear time solvable on graphs with bounded treewidth, given with a suitable tree-decomposition. (The latter can be found in $O(n \log n)$ time [8, 50].) See, amongst others, [1, 4, 5, 6, 11, 14, 29, 32, 33, 54, 59]. The underlying technique of all these results is — in a certain sense — the same, and can be described as follows: for each node of a rooted tree-representation of the input graph, some information of a certain type is computed. This computation for a node can be done quickly when given the information, computed for the children of the node. In many cases, this information is an element, taken from a finite set. In such a case, we call the problem 'finite state'. By theorem B, such an algorithm is not possible for k–Perfect Phylogeny for $k \geq 4$. For problems, that like k-Perfect Phylogeny for fixed k have no growing parameter associated with it, all general techniques to solve them on graphs with a given tree-decomposition of constant bounded treewidth can be seen as special cases of this finite state concept. (In contrast, problems like Independent Set, with a growing parameter associated to it, require a generalization of the finite state concept. Here, the 'information' is a constant size table, with each entry an integer. However, an extension of our arguments show that such approaches also cannot yield linear time algorithms.) (The result also shows, that the graph reduction method from [3] will not work for the problem with $k \geq 4$.)

In contrast, for $k = 2, 3$, k-Perfect Phylogeny is finite state. (For $k = 2$, this is trivial. For $k = 3$, it follows from the characterization in [7] that the problem can be formulated in monadic second order form, and hence, by the result of Courcelle [14], it is finite state.)

Since a standard tool for molecular biologists involves checking small subsets of characters for perfect compatibility, efficient algorithms for small k can be of use.

2 Preliminary definitions and results

A *clique* in a graph $G = (V, E)$ is a subset S of V, such that for all $v, w \in S, (v, w) \in E$. A graph $g = (V, E)$ is triangulated, if and only if it does not contain an induced cycle of length at least four. It is known [52, 26] that a graph G is triangulated if and only if there exists an linear ordering of the vertex set $v_1, v_2, \ldots, v_n$, such that for each i, the neighbors of v_i which follow v_i in the ordering, form a clique. Such an ordering is called a *perfect elimination scheme*.

The following lemma is due to Dirac [15].

Lemma 1 *Let $G = (V, E)$ be a triangulated graph which is not a complete graph. Then V contains two non-adjacent simplicial vertices.*

A graph $G = (V, E)$ with vertex coloring $c : V \to Z$ is c-triangulatable if there exists a supergraph $G' = (V, E')$, $E \subset E'$, which is properly colored by c (thus $(v, w) \in E'$ implies $c(v) \neq c(w)$) and which is triangulated. The supergraph G' is said to be a c-triangulation of G.

A useful characterization of c-triangulable graphs is with the help of tree-decompositions.

Definition A tree-decomposition of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$ with $\{X_i \mid i \in I\}$ a collection of subsets of V, and T a tree, such that

- $\bigcup_{i \in I} X_i = V$.
- For all $(v, w) \in E$, there exists an $i \in I$ with $v, w \in X_i$.
- For all $v \in V$, $\{i \in I \mid v \in X_i\}$ forms a connected subtree of T.

The treewidth of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph is the minimum treewidth over all possible tree-decompositions of that graph.

Consider $G = (V, E)$ with tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$. The graph $H = (V, E')$ with $(v, w) \in E' \Leftrightarrow \exists i \in I,\ v, w \in X_i$ contains G as a subgraph, has the same treewidth as G, and is triangulated. (Define $T_v = \{i \in I \mid v \in X_i\}$, for all $v \in V$. Then $(v, w) \in E'$, if and only if $T_v \cap T_w \neq \emptyset$. So H is the intersection graph of subtrees of a tree, hence H is triangulated, see [26].) The following proposition can now easily be observed.

Proposition 1 *A graph $G = (V, E)$ with coloring $c : V \to C$, (C a set of colors), is c-triangulatable, if and only if there exists a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of G, such that for all $i \in I$, $v, w \in V$: if $v \neq w$, and $v, w \in X_i$, then $c(v) \neq c(w)$.*

In [9] a short proof of the following fact can be found:

Proposition 2 *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$. Let $W \subset V$ form a clique in G. Then there exists an $i \in I$ with $W \subseteq X_i$.*

One can also easily verify the following propositions.

Proposition 3 *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$. Let $i_0 \in I$ be such that X_{i_0} is not a separator of G, i.e., $G[V - X_{i_0}]$ is connected. Then there exists a set $I' \subseteq I$, such that $(\{X_i \mid i \in I'\}, T[I'])$ is a tree-decomposition of G, and i_0 is a leaf of T.*

Proposition 4 *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$. Suppose $x_1, x_2, \ldots, x_r$ form a path in G, $x_1 \in X_{i_0}$, $x_2 \in X_{i_1}$. Then for every $i_2 \in I$, that lies on the path from i_0 to i_1 in T: $X_{i_2} \cap \{x_1, x_2, \ldots, x_r\} \neq \emptyset$.*

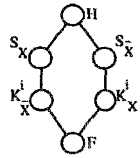

Figure 1: The Decision Component

3 Perfect Phylogeny is NP-complete

This section is devoted to the proof of the following result:

Theorem 1 *Triangulating Colored Graphs is NP-Complete, even when every color is given to exactly two vertices.*

As TCG and PP are polynomial equivalent, it directly follows from this result that Perfect Phylogeny is NP-complete.

That Triangulating Colored Graphs is in NP is obvious: given the colored graph $G = (V, E)$, present a triangulation $G' = (V, E')$, and in polynomial time we can check that G' is properly colored and triangulated.

We now show that TCG is NP-hard, by a reduction from 3-SAT to TCG.

For a given instance I of 3-SAT, we create a graph, G_I, which consists of decision components and clause components. We assume that no clause contains both a variable and its complement. For each variable X and for each clause i containing either X or $\overline{X}$, we have the decision component given by figure 1.

We call the variable H the *head*, F is called the *foot*, the variables S_X and $S_{\overline{X}}$ are called the *shoulders*, and K^i_X and $K^i_{\overline{X}}$ are called *knees*. For each variable X, we will superimpose the r copies of the decision component (corresponding to the r clauses containing X or $\overline{X}$), so that only K^i_X and $K^i_{\overline{X}}$ are not identified with other vertices. Thus, there will be one vertex H, one vertex F, and for every variable X, if X or $\overline{X}$ appear in r clauses $i_1, i_2, \ldots, i_r$, then there will be one pair of shoulders S_X and $S_{\overline{X}}$, and r pairs of knees, $K^{i_1}_X$, $K^{i_1}_{\overline{X}}$, $K^{i_2}_X$, $K^{i_2}_{\overline{X}}, \ldots, K^{i_r}_X$, $K^{i_r}_{\overline{X}}$.

We assign colors so that every color class consists of exactly two vertices. The head H and foot F are given the same color, each pair of shoulders, S_X and $S_{\overline{X}}$ is given the same color, and each pair of knees K^i_X and $K^i_{\overline{X}}$ is given the same color.

Note that there are exactly two color-respecting triangulations for the variable component for X: you either add the edges in all paths $H - K^i_X - S_X - F$, or you add the edges in all paths $H - K^i_{\overline{X}} - S_{\overline{X}} - F$. Each way of triangulating the graph can be described as adding a *Mark of Zorro* in one of two possible orientations. Thus, a triangulation either includes all edges (H, K^i_X) or all edges $(H, K^i_{\overline{X}})$. We will refer to the first orientation as the *positive orientation*, and the second as the *negative orientation*. When the triangulation is positively oriented, we will set X to *true*, and otherwise we will set X to *false*.

We now describe the clause components. For the i^{th} clause (X, Y, Z) we have the graph given by Figure 2.

Note that we do not add any new vertices, but only add edges between knees which already exist. The knees K^i_X, K^i_Y, and K^i_Z are said to be *active*, while the complements $K^i_{\overline{X}}, K^i_{\overline{Y}}$ and $K^i_{\overline{Z}}$ are said to be *inactive*. In general, if the literal L appears in the i^{th} clause, then K^i_L is said to be active, and its complement $K^i_{\overline{L}}$ is said to be inactive. Thus, for each pair of knees K^i_L and $K^i_{\overline{L}}$, exactly one will be active, and the other inactive.

As G_I can be constructed in polynomial time, given I, NP-hardness of TCG follows from the following lemma.

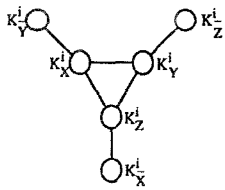

Figure 2: The Clause Component

Lemma 2 *The 3-SAT instance I is satisfiable if and only if G_I can be triangulated without introducing edges between vertices of the same color.*

Proof: We first show that if G_I has a color-respecting triangulation, then I can be satisfied. So let us assume that G_1 is a color-respecting triangulation of G_I. As we mentioned before, G_1 defines for us a truth assignment for the variables. We need to show that under this truth assignment each clause contains at least one literal, set to true.

Suppose that the truth function we derive from G_1 does not satisfy the clause $i = (X, Y, Z)$ in I; i.e. we assume that the graph G_1 does not contain any of the edges between H and K^i_X, K^i_Y, or K^i_Z. We will show that this contradicts G_1 being both properly colored by c and triangulated.

By our comments earlier, G_1 must contain the Mark of Zorro in one of the two possible orientations; since we exclude the edges (H, K^i_α), for $\alpha \in \{X, Y, Z\}$, it must include the negative orientations of the Mark of Zorro in the decision components for X, Y and Z. Thus, we assume that each of the following edges is in G_1: $(H, K^i_{\overline{\alpha}}), (S_{\overline{\alpha}}, K^i_{\overline{\alpha}}), (S_{\overline{\alpha}}, F)$, for each variable $\alpha \in \{X, Y, Z\}$ and clause i containing α. Consider the subgraph G_2 of G_1 induced by the vertex set $\{H, F, S_{\overline{\alpha}}, K^i_\alpha, K^i_{\overline{\alpha}} : \alpha$ in $\{X, Y, Z\}\}$. This subgraph G_2 is triangulated, since G_1 is triangulated. However, we will show that G_2 does not admit a perfect elimination scheme (which respects the coloring), and hence is not triangulated.

Since G_2 is triangulated, it must contain at least two simplicial vertices (see Lemma 1). Because G_2 is properly colored, only H can possibly be simplicial (every other vertex is adjacent to two vertices of the same color). Therefore, we see that G_2 can not be both triangulated and properly colored, contradicting our hypothesis.

Thus, we have shown that a color-respecting triangulation of G_I implies satisfiability of I.

We now show the converse. Suppose I is satisfiable, and that G_I is the graph we derive from I, and that f is a satisfying truth assignment for I. We will show that we can triangulate G_I without adding edges between vertices of the same color, using the truth assignment f.

We will assume that we have renamed the variables so that X is always true, and $\overline{X}$ always false. We now describe some terminology we use in defining the triangulated supergraph of G_I. Recall that we distinguish between *active* and *inactive* knees (see our discussion following the definition of the clause component). We now describe another way of distinguishing vertices. If variable X is true, we call S_X and K^i_X *true*, thus each S_X is a *true shoulder*, and each K^i_X is a *true knee*. Similarly, the complements are called *false shoulders* or *false knees*.

To triangulate G_I, add the following edges: the positively oriented Mark of Zorro in each decision component. the complete graph on $\{$*true shoulders, true knees*$\}$, and the complete bipartite graph on $\{$*true shoulders, false knees*$\}$.

Thus, we have added to the neighbor set of each true shoulder the foot F, the head H, and every knee and every true shoulder. We have added to the neighbor set of each true knee the head H and every true knee as well. It is obvious that this enlarged graph G' is properly colored. We will now show it is triangulated by exhibiting a perfect elimination scheme for G'.

Consider the following partition of the vertex set of G_I into five subsets: $S_1 = \{$ False shoulders, inactive false knees$\}$, $S_2 = \{$ The head $H\}$, $S_3 = \{$ Active false knees adjacent to inactive false knees}, $S_4 = \{$ Active false knees adjacent to inactive true knees$\}$, and $S_5 = \{$ True knees K_X^i, true shoulders S_X, and foot F $\}$.

These sets constitute a partition of the vertices of G_I into five pair-wise disjoint sets. We use these sets to produce a perfect elimination scheme, by first listing all the vertices in S_1, then those in S_2, and so forth, down to S_5. A tedious, but not very complicated case analysis shows that every vertex in simplicial in the graph which remains after the previous vertices have been deleted. We omit this analysis from this version of the paper. It now follows that G' is a properly colored triangulated supergraph of G_I. ∎

4 Non-cutset-regularity of the problem with four colors

In [20], Fellows and Abrahamson developed the theory of cutset regularity of graphs. To describe the theory, we first define some terminology used in it.

A t-boundary graph G contains a distinguished ordered subset of t nodes, called $bd(G)$). The binary operator $\oplus$ on two t-boundary graphs is defined as follows: $G \oplus H$ is the t-boundaried graph obtained by identifying the i^{th} boundary nodes in $bd(G)$ with the i^{th} boundary node in $bd(H)$, for each $i = 1, 2, \ldots, t$. For a fixed graph family F, we then define an equivalence relation on the set of t-boundary graphs as follows: Two t-boundary graphs X and Y are equivalent ($X \sim_F Y$) if and only if for every t-boundary graph Z, $X \oplus Z \in F \iff Y \oplus Z \in F$. The "small" universe U^t_{small} is defined to be the set of all t-boundaried graphs that arise in the parsing of graphs of treewidth at most t. A graph family F is *t-cutset regular* iff $\sim_F$ has finite index on U^t_{small}.

One of the main results in [20] is the following:

Theorem 2 *(Fellows and Abrahamson [20]) A graph family F is t-finite state if and only if F is t-cutset regular.*

An important consequence of this result is that, if a graph family is t-finite state, then recognition of this family can be done efficiently, and without computing obstruction sets.

Using this theorem, we can show that the class of triangulatable t-colored graphs is not t-finite state, for $t \geq 4$.

Consider the following two classes of 4-colored 4-boundary graphs:

For $r \geq 2$, let $G_r = (V_r, E_r, B, f)$ with

- $V_r = \{w_1, w_2, w_3, w_4\} \cup \{z_j \mid 1 \leq j \leq 4r\}$,
- $E_r = \{(w_1, z_1),\ (w_2, z_2),\ (w_1, z_2),\ (w_2, z_1),\ (w_3, z_{4r-1}),\ (w_4, z_{4r}),\ (w_3, z_{4r}),\ (w_4, z_{4r-1})\} \cup \{z_j, z_{j+1}) \mid 1 \leq j < 4r\}$,
- $B = \{w_1, w_2, w_3, w_4\}$, and
- $f(w_j) = j (1 \leq j \leq 4)$.

Let $c : V_r \to \{1, 2, 3, 4\}$ be the coloring of G_r, defined by $c(w_j) = j$ $(1 \leq j \leq 4)$, and $c(z_j) = (j + 1) \bmod 4 + 1$ $(1 \leq j \leq 4r)$. See Figure 3 for an example.

For $s \geq 2$, let $H_s = (V'_s, E'_s, B'_s, f'_s)$ with

- $V'_s = \{y_j \mid 1 \leq j \leq 4s\}$,
- $E'_s = \{(y_{j_1}, y_{j_2}) \mid 1 \leq j_1, j_2 \leq 4s,\ j_1 \neq j_2,\ |j_1 - j_2| \leq 2\}$,
- $B'_s = \{y_1, y_2, y_{4s-1}, y_{4s}\}$, and
- $f'_s(y_1) = 1,\ f'_s(y_2) = 2,\ f'_s(y_{4s-1}) = 3,\ f'_s(y_{4s}) = 4$.

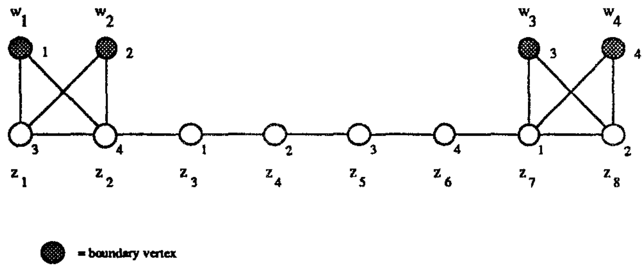

Figure 3: G_2

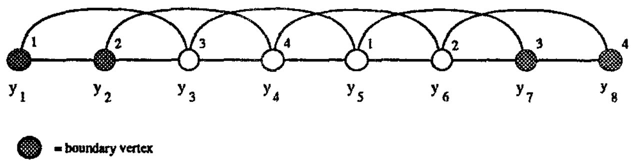

Figure 4: H_2

Let $c : V'_s \rightarrow \{1,2,3,4\}$ be the coloring of G_s, defined by $c(y_j) = (j-1) \bmod 4 + 1$. See Figure 4 for an example.

Note that for every $r, s \geq 2$, c is a coloring of $G_r \oplus H_s$.

Lemma 3 *If $s \leq r$, then $G_r \oplus H_s$ is c-triangulatable.*

Proof: First add an edge between z_2 and $z_{4(r-s)+3}$. (If $r = s$, then omit this step.) We now triangulate the cycle on the edge $(z_2, z_{4(r-s)+3})$, and the remainder of the graph independently. The cycle with edges $(z_2, z_{4(r-s)+3})$, and (z_i, z_{i+1}) for $1 \leq i < 4(r-s)+3$ can be triangulated, as the vertices on the cycle contain more than two different colors (see [30], theorem 3.1.) Further, add edges $(y_j, z_{j+4(r-s)-1})$ for all j, $2 < j \leq 4s-1$, $(y_j, z_{j+4(r-s)})$ for all j, $2 < j < 4s-1$, and $(y_j, z_{j+4(r-s)+1})$ for all j, $2 \leq j < 4s-1$. The graph now looks as depicted in Figure 6.

One can easily verify that this graph is triangulated. There are no edges between vertices of the same color. ∎

Suppose $s > r$. Let $G = G_r \oplus H_s$. Suppose we have a tree-decomposition $(X_i \mid i \in I\}, T = (I, F))$ of G, with for all $i \in I$, for all $v, w \in V$, if $v \neq w$ and $v, w \in X_i$ then $c(v) \neq c(w)$. Let H be the triangulated graph $(V, \{(v,w) \mid \exists i,\ v,w \in X_i,\ v \neq w\})$. (See Proposition 1.) From Proposition 2, it follows that there exists an $i_0 \in I$ with $X_{i_0} = \{y_1, y_2, z_1, z_2\}$, and an $i_1 \in I$ with

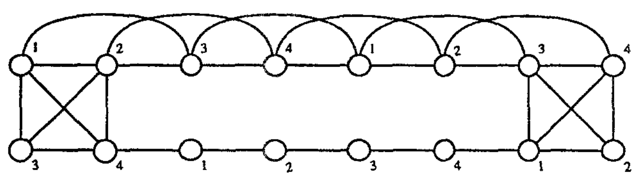

Figure 5: $G_2 \oplus H_2$

$X_{i_1} = \{y_{4s-1}, y_{4s}, z_{4r-1}, z_{4r}\}$. By Proposition 3, we may assume that i_0 and i_1 are leaves from T. Write $Y = \{y_1, y_2, \ldots, y_{4s}\}$, and $Z = \{z_1, z_2, \ldots, z_{4r}\}$.

Note from Proposition 4, that every node i_2 on the path in T between i_0 and i_1 contains at least one vertex in Z, and hence at most three vertices in Y.

Claim 1 *For every j, $1 \leq j \leq 4s-2$, there exists a node $i_2 \in I$ on the path between i_0 and i_1 in T, with $\{y_j, y_{j+1}, y_{j+2}\} \subseteq X_{i_2}$.*

Proof: Suppose the claim does not hold for certain j, $1 \leq j \leq 4s-2$. There exists a node $i_3 \in I$ with $\{y_j, y_{j+1}, y_{j+2}\} \subseteq X_{i_2}$. By assumption, i_3 lies not on the path from i_0 to i_1 in T. Let i_4 be the unique node that lies on each of the paths between i_0 and i_1, i_0 and i_2, and i_1 and i_2. There must exist a vertex $y_{j'} \notin X_{i_4}$, with $j' \in \{j, j+1, j+2\}$. Note that there exist four paths in $G[Y]$, from $\{y_j, y_{j+1}, y_{j+2}\}$ to $\{y_1, y_2, y_{4s-1}, y_{4s}\}$ that are vertex disjoint, except that two paths share the vertex $y_{j'}$. Now, by Proposition 4, X_{i_4} contains at least one vertex of each path, and, as $y_{j'} \notin X_{i_4}$, we have that $|X_{i_4} \cap Y| \geq 4$, contradiction. ∎

Note that for such i_2 on the path between i_0 and i_1, with $X_{i_2} \supseteq \{y_j, y_{j+1}, y_{j+2}\}$, there must be a $z_{j'} \in X_{i_2}$.

Claim 2 *Suppose i_2, i_3 lie on the path between i_0 and i_1 in T, and $X_{i_2} = \{y_{4\alpha+1}, y_{4\alpha+2}, y_{4\alpha+3}, z_{j_1}\}$, $X_{i_3} = \{y_{4\beta+1}, y_{4\beta+2}, y_{4\beta+3}, z_{j_2}\}$, $0 \leq \alpha, \beta < s$, $\alpha \neq \beta$. Then $c(z_{j_1}) = c(z_{j_2}) = 4$, and $j_1 \neq j_2$.*

Proof: By case analysis. Omitted from this extended abstract. ∎

It follows that there must be at least s different vertices z_j with $c(z_j) = 4$. So $G_r \oplus H_s$ can only be c-triangulatable, when $s \leq r$. Hence we have the following theorem.

Lemma 4 *$G_r \oplus H_s$ is c-triangulatable, if and only if $s \leq r$.*

It follows that every graph G_r must be in a different equivalence class, and hence TCG with four colors and 4-Perfect Phylogeny are not cutset-regular, and hence, by theorem 2 not finite-state. Clearly, the same results also hold for a larger number of colors or characteristics.

Theorem 3 *For every $k \geq 4$, k-Perfect Phylogeny, and Triangulating Colored Graphs with k colors are not finite state for bounded treewidth.*

With a slightly more complex, but further more or less similar construction one can show that the number of equivalence classes can be exponential in the number of vertices of the graphs involved. From this, it follows that not only the problem is not only not finite state, but also that no other linear time table based approach is possible. (For instance, consider Independent Set. As the size of independent sets is a parameter that can be $O(n)$ large, it is not finite state. However, there exists a 'table based' linear time algorithm for the problem, when restricted to graphs, given with a tree-decomposition of constant bounded treewidth (see e.g. [1].) When this situation occurs, then the number of equivalence classes is still polynomial. Hence, it cannot occur for the Perfect Phylogeny problem.)

References

[1] S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability – A survey. *BIT*, 25:2–23, 1985.

[2] S. Arnborg, D. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Alg. Discr. Meth.*, 8:277–284, 1987.

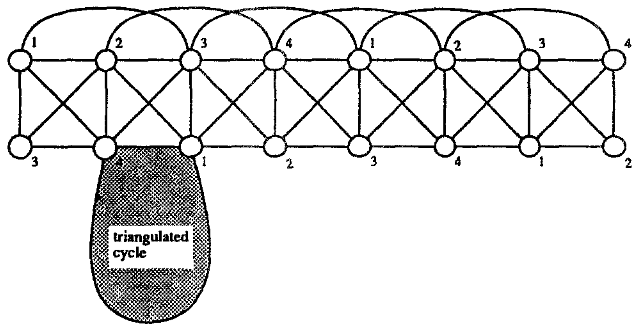

Figure 6: $G_i \oplus H_2$ triangulated

[3] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. Technical Report 90-02, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, 1990. To appear in Proceedings 4th Workshop on Graph Grammars and Their Applications to Computer Science.

[4] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs *J. Algorithms*, 12:308–340, 1991.

[5] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Disc. Appl. Math.*, 23:11–24, 1989.

[6] H. L. Bodlaender. Dynamic programming algorithms on graphs with bounded tree-width. In *Proceedings of the 15'th International Colloquium on Automata, Languages and Programming*, pages 105–119. Springer Verlag, Lecture Notes in Computer Science volume 317, 1988.

[7] H. L. Bodlaender and T. Kloks. *A simple linear time algorithm for triangulating three-colored graphs.* Technical Report RUU-CS-91-13, Department of Computer Science, Utrecht University, the Netherlands, 1991. To appear in: Proceedings STACS'92.

[8] H.L. Bodlaender and T. Kloks. Better algorithms for the pathwidth and treewidth of graphs. In *Proceedings 18'th International Colloquium on Automata, Languages and Programming*, pages 544–555. Springer Verlag, Lecture Notes in Computer Science volume 510, 1991.

[9] H. L. Bodlaender and R. H. Möhring, The pathwidth and treewidth of cographs. In *Proceedings 2nd Scandinavian Workshop on Algorithm Theory*, pages 301–309. Springer Verlag Lecture Notes in Computer Science volume 447, 1990.

[10] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comp. Syst. Sc.*, 13:335–370, 1976.

[11] R. B. Borie, R. G. Parker, and C. A. Tovey. Automatic generation of linear algorithms from predicate calculus descriptions of problems on recursive constructed graph families. Manuscript, 1988.

[12] P. Buneman. A characterization of rigid circuit graphs. *Discrete Math.* 9:205–212, 1974.

[13] J. Camin and R. Sokal, *A method for deducing branching sequences in phylogeny,* Evolution 19, (1965), pp. 311-326.

[14] B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[15] G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25: 71-76, 1961.

[16] G.F. Estabrook, *Cladistic Methodology: a discussion of the theoretical basis for the induction of evolutionary history*, Annu. Rev. Evol. Syst., 3 (1972), pp. 427-456.

[17] G.F. Estabrook, C.S. Johnson, Jr. and F.R. McMorris, *An idealized concept of the true cladistic character*, Math. Biosci. 23, 1975, pp. 263-272.

[18] G.F. Estabrook, C.S. Johnson, Jr., and F.R. McMorris, *An algebraic analysis of cladistic characters*, Discrete Math., 16, 1976, pp. 141-147.

[19] G.F. Estabrook, C.S. Johnson, Jr., and F.R. McMorris, *A mathematical foundation for the analysis of cladistic character compatibility*, Math. Biosci., 29, 1976, pp. 181-187.

[20] M. R. Fellows and K. Abrahamson, *Cutset-Regularity Beats Well-Quasi-Ordering for Bounded Treewidth.* Manuscript, Nov. 1989.

[21] J. Felsenstein. Numerical methods for inferring evolutionary trees. *The Quaterly Review of Biology*, Vol. 57, No. 4, Dec. 1982.

[22] W. M. Fitch and E. Margoliash. The construction of phylogenetic trees. *Science*, 155, 1967.

[23] L. R. Foulds, and R. L. Graham, The Steiner problem in phylogeny is NP-Complete. *Advances in Applied Mathematics,* 3:43–49, 1982.

[24] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Mathematics*, 15:835–855, 1965.

[25] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combinatorial Theory series B*, 16:47–56, 1974.

[26] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[27] D. Gusfield. *The Steiner tree problem in phylogeny.* Technical Report 332, Department of Computer Science, Yale University, Sept. 1984.

[28] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.

[29] A. Habel. *Hyperedge Replacement: Grammars and Languages.* PhD thesis, Univ. Bremen, 1988.

[30] S. Kannan and T. Warnow. Triangulating three-colored graphs. In *Proceedings Second Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 337–343, San Francisco, Jan. 1991. Also to appear in SIAM J. on Discrete Mathematics.

[31] S. Kannan and T. Warnow. Inferring evolutionary history from DNA sequences. In *Proceedings 31st Annual Symposium on the Foundations of Computer Science*, pages 362–371, St. Louis, Missouri, 1990.

[32] J. Lagergren. *Algorithms and Minimal Forbidden Minors for Tree-decomposable Graphs.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1991.

[33] C. Lautemann. Efficient algorithms on context-free graph languages. In *Proceedings of the 15th International Colloquium on Automata, Languages and Programming*, pages 362–378, 1988. Springer Verlag Lectures Notes in Computer Science volume 317.

[34] C. G. Lekkerkerker and J. Ch. Boland. Representations of a finite graph by a set of intervals on the real line, *Fund. Math.* 51:45–64, 1962.

[35] W. J. LeQuesne. The uniquely evolved character concept and its cladistic application, *Syst. Zool.*, 23:513–517, 1974.

[36] W. J. LeQuesne. The uniquely evolved character concept. *Syst. Zool.*, 26:218–223, 1977.

[37] W.J. LeQuesne, *A method of selection of characters in numerical taxonomy*, Syst. Zool., 18, pp. 201-205, 1969.

[38] W.J. LeQuesne, *Further studies on the uniquely derived character concept*, Syst. Zool., 21, pp. 281-288, 1972.

[39] W.J. LeQuesne, *The uniquely evolved character concept and its cladistic application*, Syst. Zool., 23, pp. 513-517, 1974.

[40] W.J. LeQuesne, *Discussion of preceeding papers*, In G.F. Estabrook (ed.), *Proc. Eighth International Conference on Numerical Taxonomy,*, pp. 416-429. W.H. Freeman, San Francisco, 1975.

[41] W.J. LeQuesne, *The uniquely evolved character concept*, Syst. Zool., 26, pp. 218-223, 1977.

[42] F. R. McMorris. Compatibility criteria for cladistic and qualitative taxonomic characters. In *Proceedings 8th Internatinal Conference on Numerical Taxonomy*, G.F. Estrabrook, ed., pp. 339-415. W.H. Freeman, San Francisco, 1975.

[43] F. R. McMorris. On the compatibility of binary qualitative taxonomic characters. *Bull. Math. Biol.*, 39:133-138, 1977.

[44] F. R. McMorris and C. A. Meacham. Partition intersection graphs. *Ars Combinatorica*, 16-B:135-138, 1983.

[45] F. R. McMorris, T. Warnow, and T. Wimer. *Triangulating colored graphs.* Submitted to Information Processing Letters.

[46] C. A. Meacham and G. F. Estabrook. Compatibility methods in systematics. *Annual Review of Ecology and Systematics*, 16:431–446, 1985.

[47] C. A. Meacham. Evaluating characters by character compatibility analysis. In: T. Duncan and T. F. Stuessy (eds.), *Cladistics: Perspectives on the estimation of evolutionary history*, pp. 152–165. Columbia Univ. Press: New York, 1984.

[48] C. A. Meacham. Theoretical and computational considerations of the compatibility of qualitative taxonomic characters. In: J. Felsenstein (ed.), *Numerical Taxonomy*, pages 304–314. NATO ASI Series, volume G1. Springer-Verlag: Berlin, Heidelberg, 1983.

[49] A. Proskurowski. Separating Subgraphs in k-trees: Cables and Caterpillars. *Discrete Math.*, 49:275–285, 1984.

[50] B. Reed. Finding approximate separators and computing treewidth quickly. Manuscript, 1992. To appear in: Proceedings of the 24'th Annual Symposium on Theory of Computing STOC'92.

[51] N. Robertson and P. D. Seymour. *Graph minors XIII: The disjoint path problem.* Manuscript, September 1986.

[52] D. J. Rose. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.*, 32:597–609, 1970.

[53] D. J. Rose. On simple characterization of k-trees. *Discrete Math.*, 7:317–322, 1974.

[54] P. Scheffler. Linear-time algorithms for NP-complete problems restricted to partial k-trees. Report R-MATH-03/87, Karl-Weierstrass-Institut Für Mathematik, Berlin, GDR, 1987.

[55] R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy.* W.H. Freeman, San Francisco, 1963.

[56] R. E. Tarjan. *Data Structures and Network Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, 1983.

[57] J. R. Walter. *Representations of Rigid Circuit Graphs.* Ph.D. thesis, Wayne State University.

[58] E. O. Wilson. A Consistency Test for Phylogenies Based upon Contemporaneous Species. *Systematic Zoology*, 14:214–220.

[59] T. V. Wimer. *Linear algorithms on k-terminal graphs.* PhD thesis, Dept. of Computer Science, Clemson University, 1987.