Acyclic Colorings and Chordalizations of Weakly Chordal Graphs

Andrew Lyons

Department of Computer Science, Dartmouth College

lyonsam@gmail.com

Abstract

We consider two coloring-related problems on graphs, both of which have direct applications and are NP-hard in the general case. The first is a decision problem in which we are given a graph along with a coloring on the vertices and asked whether the graph can be chordalized (adding edges to make the graph chordal) without violating the coloring. This is called the Chordalizing Colored Graphs problem, and is polynomially equivalent to a problem from evolutionary biology known as the perfect phylogeny problem.

In the second problem, we are given a graph (this time without a coloring) and asked to find an optimal coloring that satisfies the additional constraint that there be no bichromatic cycles. This problem, known as the Acyclic Coloring problem, has applications related to sparse derivative matrix evaluation in scientific computing.

We give constructive, polynomial-time algorithms for both of these problems when restricted to weakly chordal graphs. Specifically, we give an algorithm for Chordalizing Colored Graphs that, given a colored weakly chordal graph G, returns a chordalization if one exists and returns a bichromatic cycle in G otherwise. This algorithm is *certifying* in that it returns an easily-checked certificate for both "yes" and "no" instances. It follows that the acyclic colorings of a weakly chordal graph G are exactly the colorings of chordalizations of G. We use this result to show that the acyclic chromatic number of a weakly chordal graph is always one more than the treewidth, which allows us to use existing polynomial-time algorithms for computing the treewidth of these graphs in order to obtain the acyclic chromatic number. Moreover, we show that any algorithm that can construct treewidth-optimal chordalizations for a class of weakly chordal graphs can be made to construct an optimal acyclic coloring at an additional cost that is linear in the size of the chordalized graph.

1 Introduction

1.1 The Chordalizing Colored Graphs problem.

In biology, the ancestral relations among a set of species with a common ancestor are described by a phylogenetic tree (known as a *phylogeny*). From a computational perspective, one of the most fundamental problems in this area is the *perfect phylogeny* problem. Here we are given a set S of species and a collection of equivalence relations on S called *characters*[1] and asked to determine whether there is a tree satisfying a *convexity* property with respect to the characters, where the leaves of the tree are the elements of S and the internal nodes represent posited ancestral species. This is the question of whether or not there exists *some* tree in which, for every character, all species equivalent with respect to that character form a connected subtree. This question may be considered a precursor to questions such as whether a particular tree is the best (if at least one such tree exists) or whether there are trees that satisfy some subset of the characters (if there is no tree that satisfies all of them simultaneously). Buneman [12] showed that the perfect phylogeny problem is polynomially reducible to the following problem involving two well-studied notions in graph theory, namely graph coloring and chordal graphs. In the Chordalizing Colored Graphs (CCG) problem (also known as Triangulating Colored Graphs), we are given a graph G along with a vertex coloring ϕ and asked whether there exists a chordal supergraph (or *chordalization*) of G that respects the coloring. The question, therefore, is whether or not there exists some collection of edges with different colored endpoints that can be added to G so that the resulting graph is *chordal*, meaning that there are no induced cycles of length greater

[1] The perfect phylogeny problem is also known as the *character compatability problem.*

than three. If such a chordalization exists, we call it a *ϕ-chordalization* of G and say that G is *ϕ-chordalizable*. It has in fact been shown that perfect phylogeny and CCG are polynomially *equivalent* [35, 23]. In some sense this is a negative result, as this was one of the ways that the perfect phylogeny problem was shown to be NP-hard [4].On the other hand, the CCG formulation has been also been used to develop efficient algorithms, including linear-time algorithms for the special case when only three colors occur in G [22, 2, 21], and an algorithm by McMorris, Warnow, and Wimer [28] that runs in $O((n+m(k-2))^{k+1})$ time for a graph with n vertices, m edges, and k colors. Unfortunately, there is little hope for fixed-parameter tractability (with the number of colors as the parameter), as CCG has also been shown to be W[t]-hard for all $t \in \mathbb{N}$ [4]. In this paper we take a new approach—we develop a polynomial-time algorithm for cases when the graph has certain restricted structure, with no conditions on the coloring whatsoever.

1.2 The Acyclic Coloring problem.

We also consider the Acyclic Coloring (AC) problem, in which we are given a graph (this time without a coloring) and asked to find an optimal coloring that satisfies an additional constraint. Specifically, an *acyclic coloring* of a graph is a coloring in which every cycle uses at least three colors, i.e., a coloring without bichromatic cycles. This problem arose independently as both an object of purely mathematical interest [18] and as a model of efficient evaluation of sparse Hessian matrices in scientific computing [14]. For a graph G, the acyclic chromatic number $\chi_a(G)$ is defined analogously to the chromatic number $\chi(G)$ and satisfies $\chi_a(G) \geq \chi(G)$. As with CCG, most results on the complexity of AC are negative. The problem of finding an optimal acyclic coloring is NP-hard even when restricted to bipartite graphs [13] and $\chi_a(G)$ has been shown to be hard to approximate [16].

Though the bichromatic cycles prohibited by acyclic coloring are not required to be induced, it is easy to see that G contains a bichromatic cycle if and only if G contains an induced bichromatic cycle.

1.3 Bichromatic cycles and weakly chordal graphs

Before going any further, we take a moment to develop some intuition regarding the relationship between the CCG and AC problems. The focus is on bichromatic cycles.

Consider an instance (G, ϕ) of CCG where ϕ causes a bichromatic cycle C in G. We observe that such an instance will certainly be a "no" instance. To see that this is so, note that adding edges between vertices in C (remember, we cannot add edges between two vertices of the same color) will only create additional bichromatic cycles. But is the absence of a bichromatic cycle sufficient for a "yes" instance? The answer

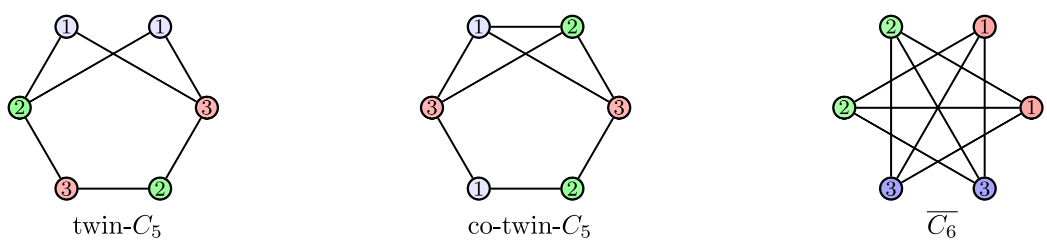

Figure 1: Three examples of graphs given with colorings ϕ. None of these graphs are ϕ-chordalizable, despite the fact that none of them contains a bichromatic cycle.

to this question is given by Figure 1, where we see three different "no" instances, none of which contain bichromatic cycles. The graphs in this figure have something in common, however, in that none of them are weakly chordal. We now define this class of graphs.

A *hole* in a graph is an induced cycle on five or more vertices[2]; an *antihole* is the complement of a hole. A graph G is *weakly chordal* (also called *weakly triangulated*) if it contains no hole or antihole. The class of weakly chordal graphs constitutes a relatively large subclass of perfect graphs that generalizes chordal graphs, permutation graphs, distance-hereditary graphs, and many other well-studied classes [9]. Weakly

[2] Note that there is not a good consensus on the use of the term "hole". For instance, the class of even-hole-free graphs does not include graphs with induced cycles of length four. We follow the usage in the book of Brandstädt, Le, and Spinrad [9].

Figure 2: Forbidden induced subgraphs for weakly chordal graphs.

chordal graphs can be recognized in $O(m^2)$ time, and efficient algorithms are known for many basic NP-hard problems when restricted to this class, including the clique problem and the traditional coloring problem [20].

1.4 Our results

In Section 2, we describe an algorithm that, given an instance (G, ϕ) of CCG, returns a ϕ-chordalization of G if one exists, and returns a bichromatic cycle in G otherwise. Note that this algorithm is *certifying* [27] in the sense that it returns an easily-checked certificate for both "yes" and "no" instances. Moreover, the fact that such an algorithm exists has the immediate consequence that a coloring ϕ of a weakly chordal graph G is an acyclic coloring if and only if it is a coloring of a chordalization of G. This means that distinguishing a "yes" instance of CCG from a "no" instance entails simply checking the given coloring for bichromatic cycles (thus a simpler algorithm may be used for the non-constructive variant of this problem, when the ϕ-chordalization of G is not required). To the author's knowledge, this is the first positive result concerning tractability of CCG for a nontrivial class of graphs.

But how does this help us for the AC problem, in which the graph is given without a coloring? In Section 3, we leverage the results on CCG mentioned above, as well as the fact that chordal graphs are perfect, to show that the acyclic chromatic number of a weakly chordal graph is always one more than the treewidth. This allows us to obtain an optimal acyclic coloring of G in the following way. We first find a chordalization H of G that is optimal with respect to treewidth (using previosuly-known existing algorithms [8]), then find an optimal coloring of H. As H is chordal, the latter step requires only linear time [17]. The details of this procedure are given in Section 3. We are also able to use this idea to obtain linear time algorithms for the acyclic coloring problem on the distance-hereditary graphs, the permutation graphs, and other well-studied subclasses of weakly chordal graphs. This is a significant generalization of the known result that AC is tractable on the class of cographs [26]. Also, though AC is NP-complete for bipartite graphs, our results show that this problem is solvable in polynomial time for the chordal bipratite graphs, which are exactly the bipartite weakly chordal graphs [9].

1.5 Terminology and notation

We consider only finite, simple, undirected graphs that are assumed without loss of generality to be connected. The vertex and edge sets of a graph G are denoted by $V(G)$ and $E(G)$. We will often use n and m to denote $|V(G)|$ and $|E(G)|$, respectively. For $S \subseteq V(G)$, we denote by $G[S]$ the subgraph of G induced by S. We denote the cycle on k vertices by C_k and its complement by $\overline{C_k}$. A *chord* in a cycle (path) is an edge between two vertices that are not consecutive in the cycle (path). A cycle or path that does not contain have any chords is said to be *chordless*. In particular, a set $C \subseteq V(G)$ forms a chordless cycle (path) in G if and only if $G[C]$ is a cycle (path). A graph is *chordal* if it has no induced cycle on four or more vertices, i.e., every cycle on four or more vertices contains a chord. A *chordalization* (or *triangulation*) H of a graph G is a chordal graph H such that $V(H) = V(G)$ and $E(H) \supseteq E(G)$.

2 Algorithms for chordalizing colored weakly chordal graphs

The goal of this section is to prove the following theorem.

Theorem 1. *There exists a polynomial-time algorithm that, given a weakly chordal graph G with a coloring ϕ, returns a ϕ-chordalization of G if one exists, and returns a bichromatic cycle in G otherwise.*

To this end, now we develop such an algorithm and argue its correctness. Our algorithm assumes that the graph G given as input is weakly chordal, and that ϕ (also part of the input) is a valid coloring of G. A run of the algorithm will consist of many recursive calls, and, since each call entails the addition of at least one edge, care is taken to ensure that no holes or antiholes are created by these additions. Moreover, we also must ensure that no bichromatic cycles are created in this way, so that any bichromatic cycle found by the algorithm is a bichromatic cycle in the original graph G. Our algorithm will either find such a bichromatic cycle (in which case the cycle will constitute a certificate of the fact that there is no ϕ-chordalization of G) or produce ϕ-chordalization of G if no such cycle is found.

We begin with a simple observation that will prove useful in limiting the types of cycles that we must consider.

Proposition 1. *Let G be a weakly chordal graph and let $uv \in E(G)$. If uv is part of a bichromatic cycle in G, then there exist vertices $t, w \in V(G)$ such that $tuvw$ induces a bichromatic cycle on four vertices in G.*

Proof. Since weakly chordal graph are hole-free, the largest induced cycle in G contains at most four vertices. it follows that any bichromatic cycle on more than four vertices in G must have a chord. Since the chord must be an edge between two vertices of different colors, the parts of the cycle on either side of the chord will again be bichromatic cycles. Continuing inductively in this fashion, we obtain that $uv \in E(G)$ is part of a bichromatic cycle on four vertices as desired. □

We now proceed with descriptions of main graph-theoretic tools, which are clique separators and two-pairs.

Clique separators. Let G be a graph, and recall that G is assumed to be connected. A set $S \subset V(G)$ is a *separator* if $G[V(G) \setminus S]$ is disconnected; a separator S is a *clique separator* if $G[S]$ is a clique. A separator S is called an *x, y-separator* if S separates x and y, i.e., if x and y are not contained in the same connected component of $G[V(G) \setminus S]$. For the sake of brevity, we will sometimes denote $G[V(G) \setminus S]$ by $G - S$.

The following easy result is certainly not new, yet it may not have appeared in the literature in this exact form. We include a proof for the sake of completeness.

Proposition 2. *If S is a clique separator of a graph G, then G is chordal if and only if $G[R \cup S]$ is chordal for every connected component R of $G - S$.*

Proof. ($\Rightarrow$): Follows immediately from the fact that being chordal is a hereditary property (i.e., every induced subgraph of a chordal graph is chordal).

($\Leftarrow$): Suppose now, for the sake of contradiction, that G contains a chordless cycle C. Since C cannot be contained entirely in S, there must be some connected component R of $G - S$ such that $C \cap R \neq \emptyset$. Let r be an arbitrary vertex in $C \cap R$, and observe that, since C cannot be contained entirely in $G[R \cup S]$, C must contain two distinct (chordless) paths from r to $G[V(G) \setminus (R \cup S)]$. However, since S separates R from the rest of the graph, these paths must encounter distinct vertices s_1, s_2 from S, respectively, immediately upon leaving R. Since $s_1, s_2 \in E(G)$, it follows that either C contains a chord or C is contained entirely within $G[R \cup S]$. In either case we have reached a contradiction, thus G contains no chordless cycle. □

Our algorithm will use this fact in the natural way: whenever G contains a clique separator S, we recursively attempt to ϕ-chordalize the graphs $G[R \cup S]$ for every component R of $G - S$. If each $G[R \cup S]$ can be successfully ϕ-chordalized, then they can be combined together to form a ϕ-chordalization of G.

Two-pairs. A pair $\{x, y\}$ of distinct, non-adjacent vertices is a *two-pair* if every induced path from x to y consists of exactly two edges. The following result, due to Hayward, Hoàng, and Maffray, demonstrates the close relationship between two-pairs and weakly chordal graphs.

Theorem 2 ([19]). *If G is a weakly chordal graph, then every induced subgraph of G that is not a clique contains a two-pair.*

Observe that $N(x) \cap N(y)$ is an x, y-separator whenever $\{x, y\}$ is a two-pair; we will often make use of this fact in our proofs. We are now ready to state the lemma that forms the crux of our algorithm.

Lemma 1. *Let G be a graph and let ϕ be a coloring of G. If $\{x, y\}$ is a two-pair in G, then x and y are contained in some bichromatic cycle in G if and only if $\phi(x) = \phi(y)$ and there exist distinct $u, v \in N(x) \cap N(y)$ such that $\phi(u) = \phi(v)$.*

Proof. ($\Leftarrow$:) This direction is immediate: $uxvy$ is a bichromatic cycle in G.

($\Rightarrow$:) Now suppose there exists a bichromatic cycle $C \subseteq V(G)$ such that $x, y \in C$. We may assume without loss of generality that C is an induced (chordless) cycle. Let $S = N(x) \cap N(y)$, and recall that S is an x, y-separator. It follows that there must exist distinct vertices $u, v \in S$ such that $u, v \in C$. Because $ux, uy, vx, vy \in E(G)$, we have $\phi(x), \phi(y) \neq \phi(u)$, thus for C to be bichromatic we must have $\phi(x) = \phi(y)$. Similarly, we have $uv \notin E(G)$ (because C is chordless) and $\phi(u), \phi(v) \neq \phi(x)$, thus $\phi(u) = \phi(v)$, which completes the proof. □

Although Lemma 1 holds for any graph G with two-pair $\{x, y\}$, we will need the fact that G is weakly chordal to ensure that such a two-pair can always be found. If x and y are found to be contained in a bichromatic cycle, then this cycle can be returned as a certificate of the fact that that G cannot be ϕ-triangulated. Otherwise, we must have that either $\phi(x) \neq \phi(y)$ or $\phi(u) \neq \phi(v)$ for all $u, v \in N(x) \cap N(y)$. To address the former scenario, we show in Section 2.1 that adding edge xy cannot create a bichromatic cycle and is guaranteed to result in a weakly chordal graph. To address the latter, we show in Section 2.2, that completing the shared neighborhood of x and y (in other words, turning $N(x) \cap N(y)$ into a clique) also cannot create a bichromatic cycle, and likewise will result in a weakly chordal graph. Since $N(x) \cap N(y)$ is a separator, the resulting graph will contain a clique separator, and we may proceed by applying Proposition 2.

2.1 Connecting two-pairs

In this section we address the scenario where $\phi(x) \neq \phi(y)$ for a two-pair $\{x, y\}$. Let $G + xy$ denote the graph obtained by adding edge xy to G. The following lemma is due to Spinrad and Sritharan.

Lemma 2 ([34])**.** *If $\{x, y\}$ is a two-pair in a graph G, then G is weakly chordal if and only if $G + xy$ is weakly chordal.*

So we know that connecting a two-pair in a weakly chordal graph results in a weakly chordal graph. The following lemma shows that connecting a two-pair with $\phi(x) \neq \phi(y)$ cannot create a bichromatic cycle. We note that the lemma holds even when G is not weakly chordal.

Lemma 3. *Let G be an arbitrary graph with two-pair $\{x, y\}$ and let let ϕ be a coloring of G such that $\phi(x) \neq \phi(y)$. For all $C \subseteq V(G)$, C is a bichromatic cycle in $G + xy$ if and only if C is a bichromatic cycle in G.*

Proof. ($\Leftarrow$:) This direction follows from the observation that a bichromatic cycle cannot be destroyed by simply adding edges.

($\Rightarrow$:) Now suppose $G + xy$ contains a bichromatic cycle C such that C is not a bichromatic cycle in G. It follows that $x, y \in C$, and there must exist $w, z \in C$ such that $w \in N(x) \setminus N(y)$ and $z \in N(y) \setminus N(x)$ (as shown in Figure 3). Recall that $N(x) \cap N(y)$ is a x, y-separator, thus $wz \notin E(G)$ and any path from w to z (as must be present in C) must contain a vertex $v \in N(x) \cap N(y)$. This, however, is a contradiction, as $\phi(z) \neq \phi(x), \phi(y)$, whereas C is supposed to be bichromatic. It follows that any bichromatic cycle in $G + xy$ must also be a bichromatic cycle in G. □

2.2 Completing the shared neighborhood of a two-pair

Lemma 4. *If G is a weakly chordal graph with two-pair $\{x, y\}$, then the graph obtained by turning $N(x) \cap N(y)$ into a clique is weakly chordal.*

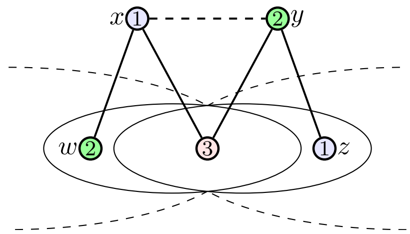

Figure 3: (Proof of Lemma 3.) The addition of edge xy cannot create a bichromatic cycle.

Proof. Let $S = N(x) \cap N(y)$ and let G' denote the graph obtained by turning S into a clique.

Suppose G' contains a hole C. Since G is hole-free, C must contain at least one edge connecting two vertices in S. However, S is a clique in G', so we must have that $C \cap S = \{u, v\}$ for distinct $u, v \in V(G)$ such that $uv \notin E(G)$. It follows that C induces a path on at least five vertices in G. Observe that $C \setminus \{u, v\}$ is disjoint from S and connected, which means $C \setminus \{u, v\}$ must be entirely contained in a single connected component of $G - S$. Let X and Y denote the connected components of $G - S$ that contain x and y, respectively. Because S is an x, y-separator, C must either be disjoint from X or Y (or both). Assume without loss of generality that C is disjoint from Y. It follows that $C \cap N(y) = \{u, v\}$, and thus $G[C \cup \{y\}]$ is a hole in G, which contradicts the fact that G is weakly chordal. We may, therefore, conclude that G' cannot contain a hole.

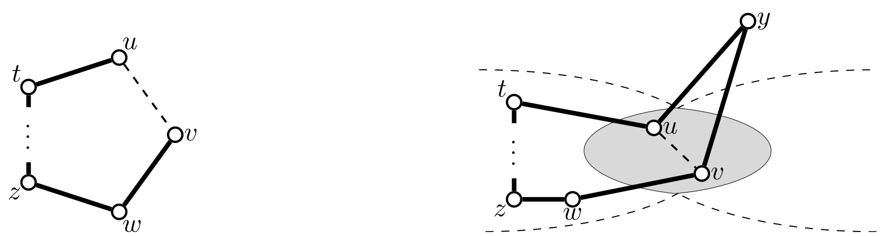

Figure 4: (Proof of Lemma 4.) The addition of edge uv cannot create a hole.

Now suppose G' contains an antihole A. We have already shown that G' contains no $C_5 = \overline{C_5}$, so A must consist of at least six vertices. Since G is antihole-free, A must result from the newly added edges among vertices in S, hence A must contain distinct vertices $u, v \in V(G)$ such that $uv \in E(G')$ and $uv \notin E(G)$.

Claim. *If uv is an edge in antihole $A \subset V(G')$ with $|A| \geq 6$, then there exist distinct $w, t \in A$ such that $vt, tw, wu \in E(G')$ and $tu, vw \notin E(G')$.*

Proof of claim. Let $ua_1a_2 \cdots a_kvb_1b_2 \cdots b_\ell$ be an ordering of A that is consistent with the cycle formed by A in $\overline{G'}$. Since $|A| \geq 6$, we have $k + \ell \geq 4$. Moreover, since $uv \in E(G')$, we have that $k, \ell \geq 1$ (meaning u and v are not adjacent in the cycle). We distinguish three cases, each depicted in Figure 5, prescribing choices for t and w in each case that have the desired properties.
Case 1: If $k, \ell > 1$, set $t = a_1$ and $w = b_1$.
Case 2: If $k = 1$ (then $\ell \geq 3$), set $t = b_\ell$ and $w = b_1$.
Case 3: If $\ell = 1$ (then $k \geq 3$), set $t = a_1$ and $w = a_k$. □

Let t, w be chosen as in the above claim. It follows that $t, w \notin S$ (because S is a clique in G'), thus $vt, tw, wu \in E(G)$ and $tu, vw, uv \notin E(G)$ (in other words, $uwtv$ induces a chordless path on four vertices in G). Moreover, because they are adjacent, t and w must lie in the same connected component of $G - S$. As in the proof of the other direction, either $t, w \notin X$ (meaning $t, w \notin N(x)$) or $t, w \notin Y$ (meaning $t, w \notin N(y)$). We may therefore assume without loss of generality that the latter is true, thus $t, w \notin N(y)$. It follows that $uwtvy$ induces a hole in G (as shown in Figure 5), which contradicts the fact that G is weakly chordal.

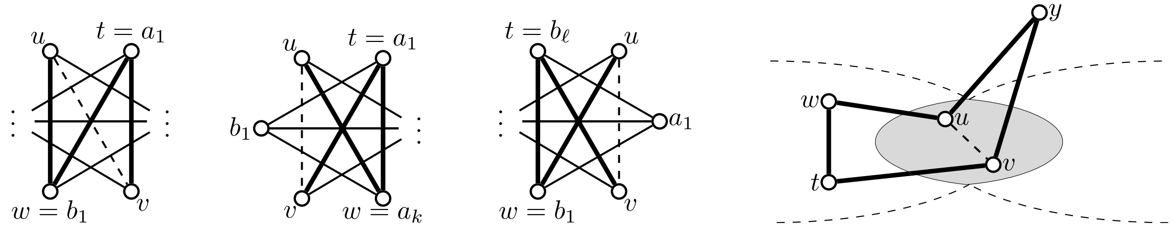

Figure 5: (Proof of Lemma 4.) If the addition of edge uv creates an antihole, then G must not have been weakly chordal to begin with.

We have shown that G' can contain neither a hole nor an antihole. It follows that G' is weakly chordal, which completes the proof of the lemma. □

We now show that completing the shared neighborhood of a two-pair $\{x, y\}$ cannot create a bichromatic cycle. We denote by $G + \text{clique}(S)$ the graph obtained by turning $S = N(x) \cap N(y)$ into a clique. The following lemma is analogous to Lemma 3; though in this case we require that G is weakly chordal.

Lemma 5. *Let G be a a weakly chordal graph with two-pair $\{x, y\}$ and let ϕ be a coloring of G such that $\phi(u) \neq \phi(v)$ for all $u, v \in S = N(x) \cap N(y)$. For all $C \subseteq V(G)$, C is a bichromatic cycle in $G + \text{clique}(S)$ if and only if C is a bichromatic cycle in G.*

Proof. (⇐:) As in Lemma 3, we observe that a bichromatic cycle cannot be destroyed by simply adding edges.

(⇒:) Now let C be an arbitrary bichromatic cycle in $G + \text{clique}(S)$ and suppose for the sake of contradiction that C is not a bichromatic cycle in G. We have that $|C \cap S| \geq 2$ (as otherwise C would also be a bichromatic cycle in G), and $|C \cap S| < 3$ (as otherwise C would contain a triangle in $G + \text{clique}(S)$). It follows, then, that $C \cap S = \{u, v\}$ for distinct $u, v \in S$ such that $uv \in E(G + \text{clique}(S)) \setminus E(G)$, and thus C induces a path in G. We again use the fact that S is an x, y-separator, observing that $C \setminus \{u, v\}$ is connected and lies in a single component of $G - S$. Thus we can assume that, without loss of generality, $C \setminus \{u, v\}$ and y lie in different connected components of $G - S$ (otherwise choose x) and in particular $C \cap N(y) = \{u, v\}$. This scenario is depicted in Figure 6. Since $|C| \geq 4$, it follows that $C \cup \{y\}$ induces a hole in G, which contradicts the fact that G is weakly chordal. Thus we conclude that any bichromatic cycle in $G + \text{clique}(S)$ is a bichromatic cycle in G, as desired. □

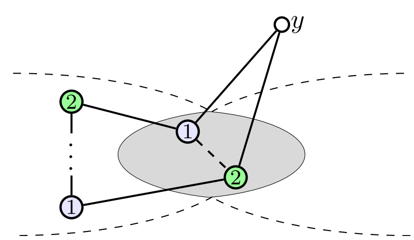

Figure 6: (Proof of Lemma 5).

2.3 The (constructive) algorithm

Our algorithm for CCG, called `TCWCG`, is shown in Algorithm 1. Note that $G + \text{clique}\big(N(x) \cap N(y)\big)$ denotes the graph obtained by turning $N(x) \cap N(y)$ into a clique for two-pair $\{x, y\}$.

Lemma 6. *Algorithm 1 is correct.*

input : weakly chordal graph G with coloring ϕ
output: a ϕ-chordalization of G if one exists, and a bichromatic cycle in G otherwise

```
if G is chordal then
    return G
find two-pair {x, y}
if φ(x) ≠ φ(y) then
    return TCWCG(G + xy, φ)
else if φ(u) ≠ φ(v) for all distinct u, v ∈ N(x) ∩ N(y) then
    G ← G + clique(N(x) ∩ N(y))
    forall the connected components A of G − S do
        G ← G ∪ TCWCG(G[A ∪ S], φ)
    return G
else
    return bichromatic cycle uxvy in G
```

Algorithm 1: TCWCG(G, ϕ)

Proof. Correctness follows from the fact that, by Lemma 5 and Lemma 3, none of the edges added by the algorithm can create a bichromatic cycle. It follows that any bichromatic cycle that is discovered and returned by the algorithm must also be a bichromatic cycle in G. If no such bichromatic cycle is found, then the algorithm continues (and, in particular, continues to add edges) until a ϕ-chordalization of G is obtained. □

Lemma 7. *Algorithm 1 can be implemented to run in polynomial time.*

Proof. We first argue that the algorithm will make at most $O(n^2)$ recursive calls. Observe that every recursive call (except those that find the input graph to be chordal) involves a two-pair, and that, for any two vertices x and y, $\{x, y\}$ will be act as the two-pair in a call of the algorithm at most once. This is because we will either discover a bichromatic cycle, add edge xy, or complete the shared neighborhood of x and y and recurse on graphs that contain at most one of x and y.

We now demonstrate that the work done at each recursive call can be accomplished in polynomial time. A call of the algorithm begins by testing whether G is chordal, which can be done in $O(n + m)$ time [17]. If the answer is no, the algorithm finds a two-pair x, y, which can be done in $O(n^{2.79})$ time [25]. We may then also decide to complete the shared neighborhood $S = N(x) \cap N(y)$ of x and y, which can be done in $O(n^2)$ time, and subsequently find the connected components of $G - S$, which can be accomplished in $O(n + m)$ time using breadth-first search.

We conclude, therefore, that each of the $O(n^2)$ recursive calls of the algorithm can be performed in polynomial time, which implies that the algorithm runs in polynomial time overall, as desired. □

Proof of Theorem 1. Follows from Lemma 6 and Lemma 7. □

2.4 A linear-time (non-constructive) algorithm

As mentioned in Section 1, Theorem 1 immediately implies the following result.

Corollary 3. *If ϕ is a coloring of a weakly chordal graph G, then G can be ϕ-chordalized if and only if ϕ is an acyclic coloring of G.*

Suppose we are given an instance of the CCG and our goal is only to determine whether the answer is "yes" or "no". it follows from Corollary 3, this non-constructive variant of CCG (in which an actual ϕ-chordalization of G is not requested) can be solved by simply checking whether ϕ is an acyclic coloring of G.

Theorem 4. *There is a linear-time algorithm that, given a weakly chordal graph G with a coloring ϕ, determines whether G can be ϕ-chordalized.*

Proof. Spinrad [33] has shown that checking a coloring for bichromatic cycles can be done in linear time. (In fact this result applies to all graphs, not just weakly chordal graphs.) □

3 Algorithms for acyclic coloring on weakly chordal graphs

We now turn our attention to the Acyclic Coloring problem. Recall that the input to this problem is a graph G given without a coloring, and the goal is to find an optimal coloring without bichromatic cycles.

We will prove the following theorem.

Theorem 5. *There exists a polynomial-time algorithm that, given a weakly chordal graph G, constructs an acyclic coloring of G that uses $\chi_a(G)$ colors.*

In fact we will prove a result that is a bit stronger, especially for certain subclasses of weakly chordal graphs, for which we obtain nearly linear-time algorithms. We first require a few definitions and some additional background.

3.1 Acyclic colorings of chordal graphs

Let $\omega(G)$ the size of the largest clique in a graph G. A graph G is *perfect* if $\chi(G') = \omega(G')$ for every induced subgraph G' of G. A classical result [17] holds that the chordal graphs are a subclass of the perfect graphs. This fact, combined with an observation regarding acyclic colorings of chordal graphs, yields the following result.

Theorem 6 ([4, 15])**.** *For every chordal graph G, $\chi_a(G) = \chi(G) = \omega(G)$.*

Proof. The rightmost equality follows from the fact that chordal graphs are perfect [17]. The leftmost equality (which could perhaps be considered folklore), follows from the fact that every coloring of a chordal graph must necessarily use at least three colors for every cycle, and is thus also an acyclic coloring. □

3.2 The connection with treewidth

The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum value of $\omega(H) - 1$ over all chordalizations H of G. We say a chordalization H of G is a *treewidth-optimal chordalization of* G if $\omega(H) - 1 = \mathrm{tw}(G)$. Treewidth is one of the most well-studied graph parameters. This is only one of a number of equivalent definitions (and it has been chosen for convenience).

Proposition 3 (Folklore)**.** *For every graph G, $\chi_a(G) \leq \mathrm{tw}(G) + 1$.*

Proof. Let H be a treewidth-optimal chordalization of G (thus $\omega(H) - 1 = \mathrm{tw}(G)$) and let ϕ be an optimal coloring of H. Since H is chordal, Theorem 6 implies that $\chi(H) = \omega(H) = \mathrm{tw}(G) + 1$. Finally, note that, because ϕ is an acyclic coloring of H, ϕ is also an acyclic coloring of G, which completes the proof. □

Note that, in general, the difference between $\chi_a(G)$ and $\mathrm{tw}(G)$ can be arbitrarily large. The planar graphs, for example, have acyclic chromatic number no greater than five [7], while their treewidth is, in general, unbounded [32].

Theorem 7. *For every weakly chordal graph G, $\chi_a(G) = \mathrm{tw}(G) + 1$.*

Proof. Let G be an arbitrary weakly chordal graph. By Proposition 3, to prove the theorem it suffices to show that $\chi_a(G) \geq \mathrm{tw}(G) + 1$. Indeed, by Corollary 3, we have that the acyclic colorings of G are exactly the colorings of chordalizations H of G. It follows that if ϕ is an acyclic coloring of G that uses $\chi_a(G)$ colors, then there exists a chordalization H of G such that ϕ is a coloring of H. Thus $\chi_a(G) \geq \chi(H) = \omega(H) \geq \mathrm{tw}(G) + 1$ as desired. □

3.3 A constructive, polynomial-time algorithm for weakly chordal graphs

Since computing $\chi_a(G)$ for a given weakly chordal graph G amounts to computing $\mathrm{tw}(G)$, we should like to know the complexity of computing $\mathrm{tw}(G)$ for a given graph. We call this the TREEWIDTH problem, and we consider an algorithm for TREEWIDTH to be constructive if it constructs a treewidth-optimal chordalization of the given graph G. This problem is NP-hard in general, though it can be determined in linear time whether G has treewidth at most k for fixed k [3] (and in fact this algorithm is constructive.) While we are not so fortunate that the weakly chordal graphs always have bounded treewidth (like the planar graphs, their treewidth is, in general, unbounded), we are able to use a result of Bouchitté and Todinca [8], who gave a constructive, polynomial-time algorithm for the TREEWIDTH problem restricted to weakly chordal graphs.

Theorem 8 ([8])**.** *There exists an algorithm that, given a weakly chordal graph G, constructs a treewidth-optimal chordalization of G in $O(n^6)$ time.*

Proof of Theorem 5. Given a weakly chordal graph G, our algorithm constructs an acyclic coloring ϕ of G that uses $\chi_a(G)$ colors. We first use the algorithm from Theorem 8 to construct a treewidth-optimal chordalization H of G. Next, we construct a coloring ϕ of H that uses $\chi(H)$ colors. Since H is chordal, such a coloring ϕ can be constructed in $O(n + |E(H)|)$ time [17]. By Corollary 7, ϕ is a acyclic coloring of G that uses $\chi(H) = \omega(H) = \mathrm{tw}(G) + 1 = \chi_a(G)$ colors, as desired. □

Notice that the additional cost of $O(n + |E(H)|)$ for constructing the coloring is very close to linear in the size of G. The algorithm of Bouchitté and Todinca, however, is quite costly. in the next section, we describe some subclasses of the weakly chordal graphs for which this costly step can be avoided.

3.4 Nearly linear-time algorithms for subclasses of weakly chordal graphs

The class of weakly chordal graphs contains many other well-studied classes. In particular, faster algorithms for TREEWIDTH exist for a number of such classes (we place particular emphasis on those graphs for which linear-time algorithms are known). The algorithm described in the proof of Theorem 5 suggests the following corollary. For definitions and further references related to these classes we again refer to the reader to the book of Brandstädt, Le, and Spinrad [9].

Corollary 9. *If $\mathcal{C}$ is a subclass of the weakly chordal graphs for which* TREEWIDTH *can be solved constructively in $f_{\mathcal{C}}(n, m)$ time for every $G \in \mathcal{C}$, then* AC *can be solved constructively in $O(f_{\mathcal{C}}(n, m) + |E(H)|)$ time for every $G \in \mathcal{C}$ where H is the constructed (treewidth-optimal) chordalization of G. In particular, there is an $O(n + |E(H)|)$ algorithm for each of the following classes of graphs:*

(i) *permutation graphs;*
(ii) *biconvex graph;*
(iii) *distance-hereditary graphs;*
(iv) *P_4-sparse graphs.*

These results (along with some others) are summarized in Figure 3.4. Each of the classes we list has the additional bonus of having recognition algorithms that are (currently) better than that of weakly chordal graphs. Note that there was already an algorithm for finding optimal acyclic colorings of cographs [26] that does not rely on first obtaining a chordalization (and thus runs in $O(n + m)$ time total).

4 Concluding remarks and an open problem

In Section 2, we proved what is arguably the central result of this paper, namely Corollary 3. This result states that, for a weakly chordal graph G, the acyclic colorings of G are exactly the colorings of chordalizations of G. It was this fact that allowed us to first obtain a certifying algorithm for CCG described in Theorem 1 and later allowed us to leverage previously-known algorithms for treewidth to solve the AC problem (Theorem 5 and Corollary 9). Given how useful this property has been, it is natural to ask which other graphs (besides the weakly chordal graphs) enjoy it.

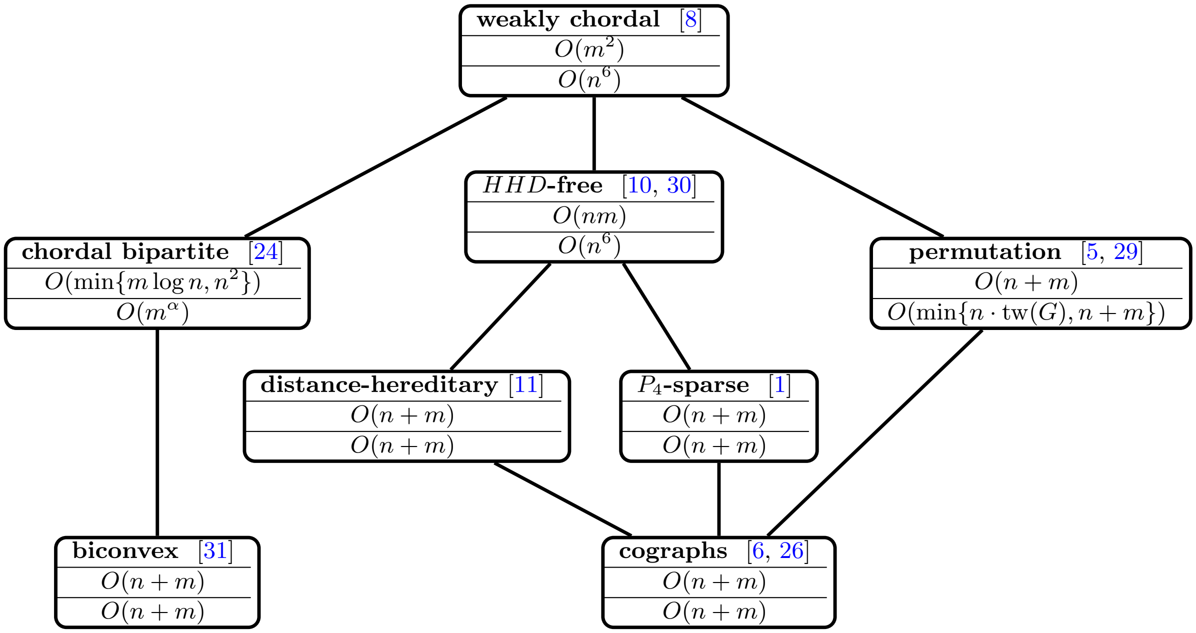

Figure 7: Some subclasses of the weakly chordal graphs along with their respective complexities of recognition (middle) and treewidth (bottom). All of the TREEWIDTH algorithms are constructive unless noted otherwise.

Problem 1. *Characterize the graphs G for which the acyclic colorings of G are exactly the colorings of chordalizations of G.*

Kannan and Warnow [22] observed that simple cycles of arbitrary length belong to this class.

Theorem 10 ([22])**.** *If G is a simple cycle on k vertices with a coloring ϕ, Then G can be ϕ-chordalized if and only if ϕ is an acyclic coloring of G.*

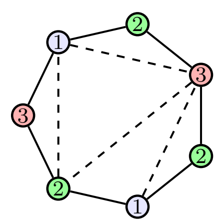

Figure 8: Kannan and Warnow showed that a single cycle of arbitrary length can be ϕ-chordalized if at least three distinct colors appear on the cycle.

Acknowledgments

The author would like to thank Kathie Cameron and Jeremy Spinrad for helpful discussions related to this work. This work was supported in part by U.S. Department of Energy, under Contract DE-AC02-06CH11357.

References

[1] L. Babel. Triangulating graphs with few P_4's. *Discrete Applied Mathematics* 89(1-3):45–57, 1998, doi:10.1016/S0166-218X(98)00115-2. 11

[2] H. Bodlaender and T. Kloks. A simple linear time algorithm for triangulating three-colored graphs. *J. Algorithms* 15(1):160–172, 1993, doi:10.1006/jagm.1993.1035. 2

[3] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* 25(6):1305–1317, 1996, doi:10.1137/S0097539793251219. 10

[4] H. L. Bodlaender, M. R. Fellows, M. T. Hallett, H. T. Wareham, and T. J. Warnow. The hardness of perfect phylogeny, feasible register assignment and other problems on thin colored graphs. *Theor. Comput. Sci.* 244(1-2):167–188, 2000, doi:10.1016/S0304-3975(98)00342-9. 2, 9

[5] H. L. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discret. Math.* 8(4):606–616, 1995, doi:10.1137/S089548019223992X. 11

[6] H. L. Bodlaender and R. H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Discret. Math.* 6(2):181–188, 1993, doi:10.1137/0406014. 11

[7] O. Borodin. On acyclic colorings of planar graphs. *Discrete Mathematics* 25:211 – 236, 1979, doi:10.1016/0012-365X(79)90077-3. 9

[8] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in of weakly triangulated graphs. *STACS*, pp. 197–206. Springer, Lecture Notes in Computer Science 1563, 1999, doi:10.1007/3-540-49116-3_18. 3, 10, 11

[9] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: A survey*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. 2, 3, 10

[10] H. Broersma, E. Dahlhaus, and T. Kloks. Algorithms for the treewidth and minimum fill-in of HHD-free graphs. *WG*, pp. 109–117. Springer, Lecture Notes in Computer Science 1335, 1997, doi:10.1007/BFb0024492. 11

[11] H. J. Broersma, E. Dahlhaus, and T. Kloks. A linear time algorithm for minimum fill-in and treewidth for distance hereditary graphs. *Discrete Applied Mathematics* 99(1-3):367–400, 2000, doi:10.1016/S0166-218X(99)00146-8. 11

[12] P. Buneman. A characterisation of rigid circuit graphs. *Discrete Mathematics* 9(3):205 – 212, 1974, doi:10.1016/0012-365X(74)90002-8. 1

[13] T. F. Coleman and J.-Y. Cai. The cyclic coloring problem and estimation of sparse Hessian matrices. *SIAM J. Alg. Disc. Meth.* 7:221–235, 1986, doi:10.1137/0607026. 2

[14] A. H. Gebremedhin, F. Manne, and A. Pothen. What color is your Jacobian? Graph coloring for computing derivatives. *SIAM Review* 47(4):629–705, 2005, doi:10.1137/S0036144504444711. 2

[15] A. H. Gebremedhin, A. Pothen, A. Tarafdar, and A. Walther. Efficient computation of sparse Hessians using coloring and automatic differentiation. *INFORMS J. Computing* 21(2):209–223, 2009, doi:10.1287/ijoc.1080.0286. 9

[16] A. H. Gebremedhin, A. Tarafdar, F. Manne, and A. Pothen. New acyclic and star coloring algorithms with application to computing Hessians. *SIAM J. Sci. Comput.* 29(3):1042–1072, 2007, doi:10.1137/050639879. 2

[17] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, vol. 57. Annals of Discrete Mathematics. Elsevier, Amsterdam, The Netherlands, second edition, 2004, doi:10.1016/0771-050X(80)90036-4. 3, 8, 9, 10

[18] B. Grünbaum. Acyclic colorings of planar graphs. *Israel J. Mathematics* 14(4):390–408, 1973, doi:10.1007/BF02764716. 2

[19] R. B. Hayward, C. T. Hoàng, and F. Maffray. Optimizing weakly triangulated graphs. *Graphs and Combinatorics* 5(1):339–349, 1989, doi:10.1007/BF01788689. 4

[20] R. B. Hayward, J. Spinrad, and R. Sritharan. Improved algorithms for weakly chordal graphs. *ACM Transactions on Algorithms* 3(2), 2007, doi:10.1145/1240233.1240237. 3

[21] R. M. Idury and A. A. Schäffer. Triangulating three-colored graphs in linear time and linear space. *SIAM Journal on Discrete Mathematics* 6(2):289–293, 1993, doi:10.1137/0406023. 2

[22] S. K. Kannan and T. J. Warnow. Triangulating 3-colored graphs. *SIAM J. Discret. Math.* 5(2):249–258, 1992, doi:10.1137/0405019. 2, 11

[23] S. K. Kannan and T. J. Warnow. Inferring evolutionary history from DNA sequences. *SIAM Journal on Computing* 23(4):713–737, 1994, doi:10.1137/S0097539791222171. 2

[24] T. Kloks and D. Kratsch. Treewidth of chordal bipartite graphs. *Journal of Algorithms* 19(2):266–281, 1995, doi:10.1006/jagm.1995.1037. 11

[25] D. Kratsch and J. Spinrad. Between $O(nm)$ and $O(n^\alpha)$. *SIAM Journal on Computing* 36(2):310–325, 2006. 8

[26] A. Lyons. Acyclic and star colorings of joins of graphs and an algorithm for cographs. *CTW*, pp. 199–202, 2009, `http://www.lix.polytechnique.fr/ctw09/ctw09-proceedings.pdf#page=211`. 3, 10, 11

[27] R. McConnell, K. Mehlhorn, S. Nher, and P. Schweitzer. Certifying algorithms. *Computer Science Review* In Press, Corrected Proof, 2010, doi:10.1016/j.cosrev.2010.09.009. 3

[28] F. R. McMorris, T. J. Warnow, and T. Wimer. Triangulating vertex-colored graphs. *SIAM Journal on Discrete Mathematics* 7(2):296–306, 1994, doi:10.1137/S0895480192229273. 2

[29] D. Meister. Treewidth and minimum fill-in on permutation graphs in linear time. *Theoretical Computer Science* 411(40-42):3685–3700, 2010, doi:10.1016/j.tcs.2010.06.017. 11

[30] S. Nikolopoulos and L. Palios. An $o(nm)$-time certifying algorithm for recognizing hhd-free graphs. *Frontiers in Algorithmics*, vol. 4613, pp. 281–292. Springer Berlin / Heidelberg, Lecture Notes in Computer Science, 2007, doi:10.1007/978-3-540-73814-5_27. 11

[31] S.-L. Peng and Y.-C. Yang. On the treewidth and pathwidth of biconvex bipartite graphs. *TAMC*, pp. 244–255. Springer, LNCS 4484, 2007, doi:10.1007/978-3-540-72504-6_22. 11

[32] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B* 36(1):49–64, 1984, doi:10.1016/0095-8956(84)90013-3. 9

[33] J. Spinrad. personal communication, 2009. 9

[34] J. Spinrad and R. Sritharan. Algorithms for weakly triangulated graphs. *Discrete Applied Mathematics* 59(2):181–191, 1995, doi:10.1016/0166-218X(93)E0161-Q. 5

[35] T. Warnow. *Combinatorial Algorithms for Constructing Phylogenetic Trees.* Ph.D. thesis, University of California, Berkeley, May 1991. 2