



Acyclic and star colorings of cographs[☆]

Andrew Lyons^{*}

Department of Computer Science, Dartmouth College, United States

ARTICLE INFO

Article history:

Received 23 September 2009

Received in revised form 30 March 2011

Accepted 7 April 2011

Available online 3 June 2011

Keywords:

Acyclic coloring

Star coloring

Cograph

Treewidth

Pathwidth

Triangulating colored graphs

ABSTRACT

An *acyclic coloring* of a graph is a proper vertex coloring such that the union of any two color classes induces a disjoint collection of trees. The more restricted notion of *star coloring* requires that the union of any two color classes induces a disjoint collection of stars. We prove that every acyclic coloring of a cograph is also a star coloring and give a linear-time algorithm for finding an optimal acyclic and star coloring of a cograph. If the graph is given in the form of a cotree, the algorithm runs in $O(n)$ time. We also show that the acyclic chromatic number, the star chromatic number, the treewidth plus 1, and the pathwidth plus 1 are all equal for cographs.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

A *proper vertex coloring* (or *proper coloring*) of a graph G is a mapping $\phi : V \rightarrow \mathbb{N}^+$ such that if a and b are adjacent vertices, then $\phi(a) \neq \phi(b)$. The chromatic number of a graph G , denoted by $\chi(G)$, is the minimum number of colors required in any proper coloring of G . An *acyclic coloring* of a graph is a proper coloring such that the subgraph induced by the union of any two color classes is a disjoint collection of trees. A *star coloring* of a graph is a proper coloring such that the subgraph induced by the union of any two color classes is a disjoint collection of stars. The acyclic and star chromatic numbers of G are defined analogously to the chromatic number and are denoted by $\chi_a(G)$ and $\chi_s(G)$, respectively. Since a disjoint collection of stars constitutes a forest, it follows that every star coloring is also an acyclic coloring and $\chi_a(G) \leq \chi_s(G)$ for every graph G . We will often find it useful to work with the alternative definitions implied by the following (folklore) observation.

Observation 1. *The following are true whenever ϕ is a proper coloring of a graph G .*

ϕ is an acyclic coloring of G if and only if every cycle in G uses at least three colors.

ϕ is a star coloring of G if and only if every path on four vertices in G uses at least three colors.

A great deal of graph-theoretical research has been conducted on acyclic and star coloring since they were introduced in the early seventies by Grünbaum [18]. Our investigation of these problems from an algorithmic point of view is motivated in part by their applications in combinatorial scientific computing, where they model the optimal evaluation of sparse Hessian matrices. In fact, these coloring problems were independently discovered and studied by the scientific computing community. The survey of Gebremedhin et al. [15] gives a history of the subject as well as an overview of the use of these coloring variants in computing sparse derivative matrices.

[☆] A preliminary version of this article, entitled “Acyclic and star colorings of joins of graphs and an algorithm for cographs”, appeared in *Proceedings of the 8th Cologne–Twente Workshop on Graphs and Combinatorial Optimization*, 2009 [23].

^{*} Corresponding address: Dartmouth College, Department of Computer Science, 6211 Hinman, 03755 Hanover, NH, United States. Fax: +1 630 646 1672. E-mail address: lyonsam@gmail.com.

The acyclic and star coloring problems are both NP-hard, and most results concerning their complexity on special classes of graphs are negative. In particular, both problems remain NP-hard even when restricted to bipartite graphs [8,9]. In addition, Albertson et al. [1] showed that the problem of determining whether the star chromatic number is at most 3 is NP-complete even for planar bipartite graphs. The authors also showed that it is NP-complete to decide whether the chromatic number of a graph G is equal to the star chromatic number of G , even if G is a planar graph with chromatic number 3. Inapproximability results for both problems are given in [16].

Researchers have obtained a few positive algorithmic results for these problems on graphs for which the acyclic or star chromatic number is bounded by a constant. In particular, Skulrattanakulchai [24] gives a linear-time algorithm for finding an acyclic coloring of a graph with maximum degree 3 that uses four colors or fewer, and Fertin and Raspaud [12] give a linear-time algorithm for finding an acyclic coloring of a graph with maximum degree 5 that uses nine colors or fewer. To our knowledge, prior to this work no polynomial-time algorithm was known for either of these problems on a nontrivial class of graphs for which the acyclic or star chromatic number is unbounded.

In this paper, we consider acyclic and star colorings of cographs. This class of graphs, which we define formally in Section 2, was discovered independently by a number of researchers, and hence has many characterizations. We refer the interested reader to the book of Brandstädt et al. [7] for additional background and details related to cographs. Many problems that are NP-complete on general graphs have polynomial-time algorithms when restricted to cographs, in part because of the nice decomposition properties that these graphs exhibit. Nevertheless, problems such as list coloring and achromatic number remain NP-complete on this class [2,20]. Our motivation, however, stems also from a new characterization of cographs: they are exactly the graphs for which every acyclic coloring is also a star coloring. We begin Section 2 with a simple proof of this fact.

Bodlaender and Möhring [5] showed that the pathwidth of a cograph equals its treewidth. In Section 3, we prove that the acyclic colorings of a cograph G coincide with the proper colorings of triangulations of G . As a consequence, we find that the acyclic chromatic number, the star chromatic number, the treewidth plus 1, and the pathwidth plus 1 are all equal for cographs. Additionally, we discuss some implications of our results for the triangulating colored graphs problem, which is related to a problem from evolutionary biology.

In Section 4, we describe an algorithm that, given a cograph G , produces an optimal acyclic and star coloring of G . When G is given as an adjacency list, our algorithm runs in $O(n+m)$ time, where n and m are the numbers of vertices and edges in G , respectively; only $O(n)$ time is required when G is given in the form of a cotree, which is a concise tree-based representation of G .

2. Cographs and cotrees

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs such that $V_1 \cap V_2 = \emptyset$. The *disjoint union* of G_1 and G_2 is the graph $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$. The *join* of G_1 and G_2 , denoted by $G_1 * G_2$, is the graph obtained by adding all possible edges between G_1 and G_2 , i.e., $G_1 * G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{v_1 v_2 \mid v_1 \in V_1, v_2 \in V_2\})$.

Definition 1 (Cograph [11]). A graph $G = (V, E)$ is a *cograph* if and only if one of the following conditions holds:

- (i) $|V| = 1$;
- (ii) there exist cographs G_1, \dots, G_k such that $G = G_1 \cup G_2 \cup \dots \cup G_k$;
- (iii) there exist cographs G_1, \dots, G_k such that $G = G_1 * G_2 * \dots * G_k$.

As noted in Section 1, this class can be characterized in a number of ways. We will find the following characterization most useful.

Theorem 1 ([7, Theorem 11.3.3]). *The cographs are exactly those graphs that do not contain an induced path on four vertices.*

Theorem 2. *The cographs are exactly those graphs G for which every acyclic coloring of G is also a star coloring of G .*

Proof. (\Rightarrow) Let ϕ be an acyclic coloring of a cograph G , and let P be a path on four vertices in G . Since, by Theorem 1, G cannot contain an induced path on four vertices, the graph induced by P must either be a cycle (in which case it uses at least three colors by Observation 1) or contain a triangle (in which case it uses at least three colors because ϕ is a proper coloring). Thus, by Observation 1, ϕ is a star coloring of G .

(\Leftarrow) Now suppose G is a graph in which every acyclic coloring is also a star coloring and assume for the sake of contradiction that G contains an induced path on vertices $abcd$ (in that order). Let ϕ be a coloring that assigns each vertex in G its own (distinct) color with the exceptions $\phi(a) = \phi(c)$ and $\phi(b) = \phi(d)$. Because $abcd$ induces a path, we have that ϕ is both proper and acyclic, yet, by Observation 1, ϕ is not a star coloring, which is a contradiction. \square

Corollary 3. *For every cograph G , $\chi_s(G) = \chi_a(G)$.*

The fact that every acyclic coloring of a cograph is also a star coloring means that, for the bulk of this paper, we may restrict our attention to acyclic colorings.

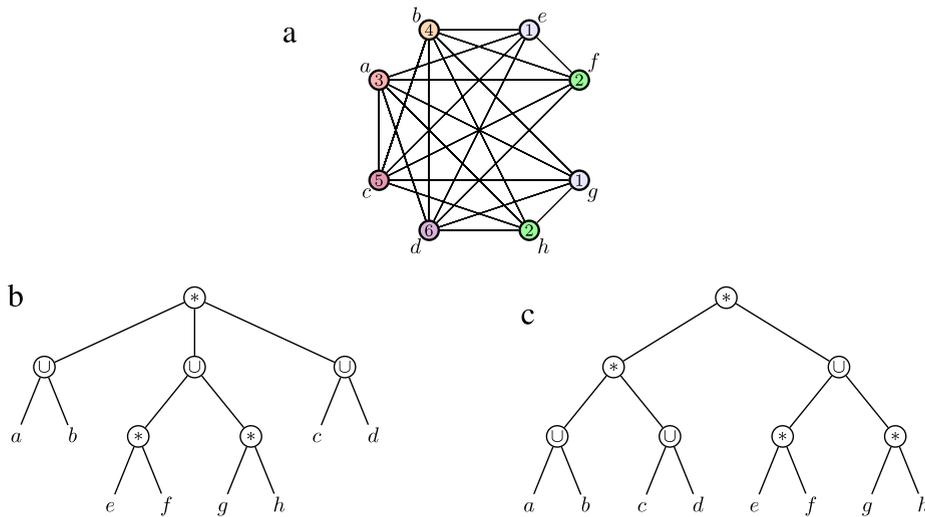


Fig. 1. (a) A cograph G ; (b) its canonical cotree; (c) a binary cotree. The graph G is shown along with an optimal acyclic coloring, which is (necessarily) also an optimal star coloring by Theorem 2.

2.1. Cotrees

Cographs can be recognized in linear time [11,19], and most recognition algorithms also produce a special decomposition structure in the same time bound when the input graph G is a cograph. We now introduce this structure, which is often used in algorithms designed to work on cographs. A *cotree* for a cograph G is a rooted tree T whose leaves correspond to the vertices of G and whose internal nodes are given labels from $\{*, \cup\}$ such that two vertices in G are adjacent if and only if the lowest common ancestor of the corresponding leaves in T is a \otimes -node. (For the sake of clarity, we will use the word “node” when referring to cotrees, whereas the term “vertex” will be reserved for the context of the original graph G .) For a node τ in T , V_τ denotes the set of vertices in G that correspond to leaves in the subtree of T rooted at τ ; we denote by G_τ the subgraph of G induced by V_τ . Cotrees thus describe the recursive construction of cographs described in Definition 1.

While there may be many cotrees for a given cograph G , the *canonical cotree* of G , which is characterized by the property that any path from a leaf to the root alternates between \cup -nodes and \otimes -nodes, is unique up to isomorphism [10]. It is often more convenient to work with cotrees whose internal nodes have exactly two children. Since the operations \cup and $*$ are commutative and associative, one can show that any cotree T can, in linear time, be converted into a cotree T' such that T' meets this condition and has size linear in that of T [5]. An example is shown in Fig. 1.

We conclude this section with a lemma that suggests a natural way that a binary cotree can be used to compute the acyclic and star chromatic numbers of a cograph; a formal description and analysis of our algorithm is given in Section 4. Note that this result applies to disjoint unions and joins of *general* graphs, and is analogous to a result of Bodlaender and Möhring [5, Lemma 3.4] concerning treewidth and pathwidth (we define these notions in the next section).

Lemma 2.1 ([23]). *The following hold for any graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.*

- (i) $\chi_a(G_1 \cup G_2) = \max\{\chi_a(G_1), \chi_a(G_2)\}$;
- (ii) $\chi_s(G_1 \cup G_2) = \max\{\chi_s(G_1), \chi_s(G_2)\}$;
- (iii) $\chi_a(G_1 * G_2) = \min\{\chi_a(G_1) + |V_2|, \chi_a(G_2) + |V_1|\}$;
- (iv) $\chi_s(G_1 * G_2) = \min\{\chi_s(G_1) + |V_2|, \chi_s(G_2) + |V_1|\}$.

Proof. The proofs of (i) and (ii) are obvious.

Our proof of (iii) begins by showing that $\chi_a(G_1 * G_2) \leq \min\{\chi_a(G_1) + |V_2|, \chi_a(G_2) + |V_1|\}$. We describe an algorithm that, given optimal acyclic colorings of G_1 and G_2 , produces an acyclic coloring ϕ of $G_1 * G_2$ that uses the desired number of colors. Let ϕ_1 and ϕ_2 be arbitrary optimal acyclic colorings of G_1 and G_2 , respectively. Assume without loss of generality that $\chi_a(G_2) + |V_1| \leq \chi_a(G_1) + |V_2|$. We construct ϕ as follows. Color those vertices in V_2 the same as they are colored by ϕ_2 , using the colors in $\{1, \dots, \chi_a(G_2)\}$. Color those vertices in V_1 such that each $v \in V_1$ receives a distinct color in $\{\chi_a(G_2) + 1, \dots, \chi_a(G_2) + |V_1|\}$. Suppose that ϕ causes a bichromatic cycle $C \subseteq V$ in $G_1 * G_2$. Since each vertex in V_1 gets a distinct color and no vertex in V_1 shares a color with a vertex in V_2 , it follows that any bichromatic cycle in $G_1 * G_2$ must be contained entirely in V_2 , which contradicts the fact that ϕ_2 is an acyclic coloring of the subgraph of $G_1 * G_2$ induced by V_2 . Thus ϕ is an acyclic coloring of $G_1 * G_2$, which completes this direction of the proof.

Now let ϕ be an optimal acyclic coloring of $G_1 * G_2$. Observe that, since every vertex from V_1 is adjacent to every vertex from V_2 , V_1 and V_2 must receive disjoint sets of colors from ϕ . Moreover, ϕ must assign $|V_1|$ distinct colors to the vertices in

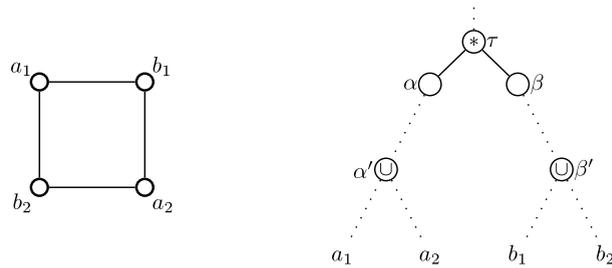


Fig. 2. The cotree structure of an induced cycle on four vertices in a cograph, as in the proof of Lemma 3.1.

V_i for some $i \in \{1, 2\}$, as otherwise there would be vertices $a_1, b_1 \in V_1$ such that $\phi(a_1) = \phi(b_1)$ and vertices $a_2, b_2 \in V_2$ such that $\phi(a_2) = \phi(b_2)$, and thus ϕ would color the cycle $a_1a_2b_1b_2$ with only two colors (contradicting the fact that ϕ is an acyclic coloring). Finally, since ϕ must be an acyclic coloring of every induced subgraph of $G_1 * G_2$, we have that ϕ must assign at least $\chi_a(G_i)$ colors to the subgraph induced by V_i for $i \in \{1, 2\}$, which implies that ϕ must use at least $\min\{\chi_a(G_1) + |V_2|, \chi_a(G_2) + |V_1|\}$ colors, which completes the proof of (iii).
 The proof of (iv) is similar to that of (iii). \square

3. Acyclic colorings and triangulations of cographs

A graph is *chordal* if it has no induced cycle on four or more vertices, i.e., every cycle of length greater than three has a chord. A *triangulation* of a graph $G = (V, E)$ is a chordal graph $G^+ = (V, E^+)$ such that $E \subseteq E^+$. The following notion allows us to characterize those cographs which are also chordal in terms of the structure of their cotrees.

Definition 2 (Skew Cotree). A cotree T is said to be *skew* if at most one child of every $\textcircled{*}$ -node in T has some \textcircled{U} -node as a descendant.

Lemma 3.1. A cograph G contains an induced cycle on four or more vertices if and only if T is not skew for every cotree T of G .

Proof. (\Rightarrow) Suppose G contains an induced cycle on four or more vertices. Since G is a cograph, and thus cannot contain an induced path on four vertices, such a cycle must consist of exactly four vertices, say $a_1b_1a_2b_2$ (in that order). Let T be an arbitrary cotree of G and denote by τ the lowest common ancestor in T of the leaves corresponding to a_1 and b_1 . Since a_1 and b_1 are adjacent, we have that τ is a $\textcircled{*}$ -node and has distinct children α and β such that α is an ancestor of a_1 and β is an ancestor of b_1 . We will demonstrate that T is not skew by showing that both α and β are ancestors of \textcircled{U} -nodes. Let α' be the lowest common ancestor in T of a_1 and a_2 . Since α' is a \textcircled{U} -node (because a_1 and a_2 are non-adjacent), α' must be distinct from τ . At the same time, α', α , and τ are all ancestors of a_1 , implying that all of these nodes lie on the unique path in T from a_1 to the root. It follows that α' must be a descendant of τ , as otherwise the lowest common ancestor of a_2 and b_1 would be α' (a \textcircled{U} -node). In particular, α' must also be a descendant of α because α is a child of τ (note that we may have $\alpha' = \alpha$). Similar reasoning with respect to the lowest common ancestor β' in T of b_1 and b_2 implies that β' is a \textcircled{U} -node and a descendant of β . It follows that T is not skew (see Fig. 2), which completes this direction of the proof.

(\Leftarrow) Now suppose there exists some cotree T of G that is not skew. It follows that T contains a $\textcircled{*}$ -node τ such that τ is the lowest common ancestor in T of two distinct \textcircled{U} -nodes which we will call α' and β' . Now let a_1 and a_2 (resp. b_1 and b_2) be any pair of leaves whose lowest common ancestor in T is α' (resp. β'). Since the lowest common ancestor in T of a_i and b_j is τ for $i, j \in \{1, 2\}$, and τ is a $\textcircled{*}$ -node, we have that $a_1b_1a_2b_2$ induces a cycle in G , which completes the proof. \square

Definition 3. Let ϕ be a proper coloring of a cograph G and let T be a cotree of G . A node τ in T is said to be *saturated* by ϕ if $\phi(x) \neq \phi(y)$ for all distinct $x, y \in V_\tau$.

Lemma 3.2. If ϕ is a proper coloring of a cograph G , then the following are equivalent.

- (i) ϕ is an acyclic coloring of G ;
- (ii) if T is a cotree of G , then for every $\textcircled{*}$ -node τ in T , at most one child of τ is not saturated by ϕ ;
- (iii) ϕ is a proper coloring of some triangulation of G .

Proof. ((i) \Rightarrow (ii)) Suppose ϕ is an acyclic coloring of G and assume for the sake of a contradiction that there exists some $\textcircled{*}$ -node τ in T with distinct children α and β such that neither α nor β is ϕ -saturated. It follows that α is an ancestor of leaves a_1 and a_2 such that $\phi(a_1) = \phi(a_2)$ and, likewise, β is an ancestor of distinct leaves b_1 and b_2 such that $\phi(b_1) = \phi(b_2)$. It follows that, with respect to ϕ , $a_1b_1a_2b_2$ induces a bichromatic cycle in G , which contradicts the fact that ϕ is an acyclic coloring of G .

((ii) \Rightarrow (iii)) Now suppose ϕ satisfies (ii) and let T be an arbitrary cotree of G . The desired triangulation of G is constructed as follows. For every node τ in T that is saturated by ϕ , turn the subgraph G_τ into a clique by changing every \textcircled{U} -node in the

subtree of T rooted at τ into a \otimes -node. By (ii), the resulting cotree T^+ is skew, and Lemma 3.1 implies that T^+ is a cotree of a triangulation G^+ of G . Since ϕ is a proper coloring of G , and none of the new edges added to form G^+ connect vertices of the same color, it follows that ϕ is a proper coloring of G^+ as desired.

((iii) \Rightarrow (i)) Suppose ϕ is a proper coloring of some triangulation G^+ of G . Since every cycle in G is also a cycle in G^+ (which is a supergraph of G), and every cycle in G^+ has a chord, it follows that ϕ must use at least three colors for every cycle in G ; thus ϕ is an acyclic coloring of G . \square

3.1. Treewidth and pathwidth of cographs

The *clique number* of a graph G , denoted by $\omega(G)$, is the largest number of pairwise adjacent vertices in G . The *treewidth* of a graph G , denoted by $\text{tw}(G)$, is the minimum value of $\omega(G^+) - 1$ over all triangulations G^+ of G .

Theorem 4 (Folklore). For every graph G , $\chi_a(G) \leq \text{tw}(G) + 1$.

Proof. By the definition of treewidth, there exists a triangulation G^+ of G such that $\omega(G^+) = \text{tw}(G) + 1$. Since G^+ is chordal and since chordal graphs are perfect [17], G^+ further satisfies $\omega(G^+) = \chi(G^+)$. The desired inequality then follows from the observation that every proper coloring of G^+ is an acyclic coloring of G (as shown in the proof of Lemma 3.2). \square

Theorem 5. For every cograph G , $\chi_a(G) = \text{tw}(G) + 1$.

Proof. By Theorem 4, it suffices to show that $\chi_a(G) \geq \text{tw}(G) + 1$. To see that this is so, consider an arbitrary optimal acyclic coloring of G which, by Lemma 3.2, is a proper coloring of some triangulation G^+ of G . The desired inequality then follows from the fact that $\chi_a(G) \geq \omega(G^+) \geq \text{tw}(G) + 1$. \square

A graph is an *interval graph* if its vertices can be put in correspondence with intervals on the real line such that two vertices are adjacent if and only if the corresponding intervals have a nonempty intersection. An *intervalization* of a graph $G = (V, E)$ is an interval graph $G^+ = (V, E^+)$ such that $E \subseteq E^+$. The *pathwidth* of a graph G , denoted by $\text{pw}(G)$, is the minimum value of $\omega(G^+) - 1$ over all intervalizations G^+ of G . Note that since the interval graphs form a proper subclass of the chordal graphs, we have that $\text{tw}(G) \leq \text{pw}(G)$ for all graphs G . Bodlaender and Möhring obtained the following result by showing that every triangulation of a cograph G is also an intervalization of G .

Theorem 6 ([5]). For every cograph G , $\text{tw}(G) = \text{pw}(G)$.

Combining Corollary 3, Theorems 5 and 6 we obtain the following result.

Corollary 7. For every cograph G , $\chi_s(G) = \chi_a(G) = \text{tw}(G) + 1 = \text{pw}(G) + 1$.

We note that Corollary 7 also follows from an inductive application of Lemma 2.1. However, the results of the next section, as well as the proof of correctness of the algorithm given in Section 4, rely on the intermediate results used in the proof that we have given here.

3.2. Triangulations of colored graphs and the perfect phylogeny problem

Let G be a graph given with a proper coloring ϕ . We say that G is ϕ -*triangulatable* if there exists a triangulation G^+ of G such that ϕ is a proper coloring of G^+ . In the general case, determining whether G is ϕ -triangulatable is NP-complete [4]. This is known as the *triangulating colored graphs* problem, which is polynomially equivalent to the perfect phylogeny problem from evolutionary biology [21]. The following result follows immediately from Lemma 3.2, which characterizes the colorings ϕ for which a cograph G is ϕ -triangulatable as exactly the acyclic colorings of G . (Note that we can check in polynomial time whether ϕ is an acyclic coloring of G , and the procedure described in the proof of Lemma 3.2 can be used to obtain a compatible triangulation in case one is desired.)

Corollary 8. There exists a polynomial-time algorithm that, given a cograph G along with a proper coloring ϕ of G , determines whether G is ϕ -triangulatable and, if so, produces a compatible triangulation.

Fig. 3 illustrates this concept for the graph depicted in Fig. 1(a).

4. The algorithm for acyclic and star coloring

In this section we prove the following theorem.

Theorem 9. There exists an algorithm that, given a binary cotree T for cograph G , produces an optimal acyclic and star coloring of G in $O(n)$ time. Moreover, the coloring obtained is also an optimal star coloring.

As mentioned in Section 2, an arbitrary cotree can be transformed into a binary cotree in $O(n)$ time. If G is given in the form of an adjacency list, we simply build a cotree of G (in $O(n + m)$ time [11]) before running the algorithm, which results in an overall running time of $O(n + m)$. We now give an informal description of the algorithm, which consists of two phases.

In the first phase, we traverse the cotree from the leaves to the root, computing for every τ in T the values $|V_\tau|$ and $\chi_a(G_\tau)$ in accordance with Lemma 2.1.

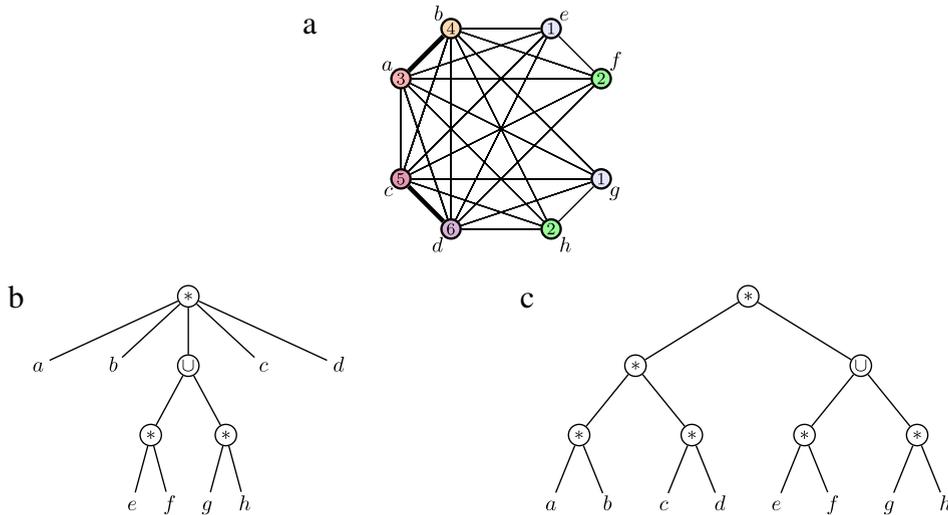


Fig. 3. (a) A triangulation G^+ of the graph G depicted in Fig. 1(a) (bold edges added during the triangulation process), which satisfies $\chi_a(G) = \chi(G^+) = \omega(G^+) = 6$; (b) the canonical cotree for G^+ ; (c) the binary cotree T^+ for G^+ obtained from the binary cotree depicted in Fig. 1(c).

The second phase, in which an optimal acyclic and star coloring ϕ is constructed, consists of a top-down traversal of the cotree. Recall that, by Lemma 2.1, for every \odot -node in T with children α and β , either α or β must be saturated by ϕ , meaning that the vertices associated with leaves in the corresponding subtree are assigned pairwise-distinct colors. In order to decide which subtree is to be saturated, the algorithm makes use of the values $|V_\alpha|$, $|V_\beta|$, $\chi_a(G_\alpha)$, and $\chi_a(G_\beta)$, which are all computed during the first phase.

4.1. Phase I: computing the quantity $\chi_a(G) = \chi_s(G)$

Data: binary cotree T for a cograph G
Result: values $|V_\tau|$ and $\chi_a(G_\tau)$ for every node $\tau \in T$
Procedure COMPUTEAC(t)
input: cotree node τ from T
begin
 if τ is a leaf **then**
 $|V_\tau| \leftarrow 1$
 $\chi_a(G_\tau) \leftarrow 1$
 else /* τ has children α and β */
 COMPUTEAC(α)
 COMPUTEAC(β)
 $|V_\tau| = |V_\alpha| + |V_\beta|$
 if t is a \odot -node **then**
 $\chi_a(G_\tau) \leftarrow \max\{\chi_a(G_\alpha), \chi_a(G_\beta)\}$
 else /* τ is a \odot -node */
 $\chi_a(G_\tau) \leftarrow \min\{|V_\alpha| + \chi_a(G_\beta), |V_\beta| + \chi_a(G_\alpha)\}$

Algorithm 1: Calling COMPUTEAC(ρ) where ρ is the root of T yields $\chi_a(G)$ in $O(n)$ time.

Recall that the initial phase of our main algorithm, which is shown in Algorithm 1, computes the quantity $\chi_a(G_\tau) = \chi_s(G_\tau)$ for every node τ in the cotree. If one desires only the quantity $\chi_a(G) = \chi_s(G)$ —but not necessarily a corresponding coloring ϕ that achieves this quantity—then Algorithm 1 is all that is required.

Lemma 4.1. Algorithm 1 is correct and runs in $O(n)$ time.

Proof. Correctness follows from an inductive application of Lemma 2.1. Trivially, $\chi_a(G_\lambda) = 1$ for every leaf λ of T . Observe that, when called on the root node ρ of T , the procedure COMPUTEAC() correctly computes the quantity $\chi_a(G_\tau) = \chi_s(G_\tau)$ for every node τ in T . The correct value of the quantity $\chi_a(G) = \chi_s(G)$ is thus obtained as $\chi_a(G_\rho)$ when COMPUTEAC() is called on the root node ρ of T .

To establish the running time, observe that the recursive calls of COMPUTEAC() are in one-to-one correspondence with the nodes in T , of which there are $O(n)$. \square

4.2. Phase II: construction of an optimal coloring

Data: binary cotree T for a cograph G ; values $|V_\tau|$ and $\chi_a(G_\tau)$ for every node τ in T

Result: acyclic coloring $\phi: V \rightarrow \{1, \dots, \chi_a(G)\}$

Procedure SATURATE(τ, K)

input: node $\tau \in T$, positive integer K

begin

if τ is a leaf **then**

$\phi(\tau) \leftarrow K$

else /* τ has children α and β */

 SATURATE(α, K)

 SATURATE($\beta, K + |V_\alpha|$)

Procedure ASSIGNCOLORS(τ, K)

input: node τ from T , positive integer K

begin

if τ is a leaf **then**

$\phi(\tau) \leftarrow K$

else /* τ has children α and β */

if τ is a \odot -node **then**

 ASSIGNCOLORS(α, K)

 ASSIGNCOLORS(β, K)

else /* τ is a \otimes -node */

if $|V_\alpha| + \chi_a(G_\beta) \leq |V_\beta| + \chi_a(G_\alpha)$ **then**

 SATURATE(α, K)

 ASSIGNCOLORS($\beta, K + |V_\alpha|$)

else /* $|V_\beta| + \chi_a(G_\alpha) < |V_\alpha| + \chi_a(G_\beta)$ */

 SATURATE(β, K)

 ASSIGNCOLORS($\alpha, K + |V_\beta|$)

Algorithm 2: Calling ASSIGNCOLORS($\rho, 1$) where ρ is the root of T yields an optimal acyclic coloring G in $O(n)$ time. The coloring obtained is also an optimal star coloring of G .

The procedures for the second phase of our algorithm, which require the values $|V_\tau|$ and $\chi_a(G_\tau)$ for every node τ in T (as computed in the first phase), are shown in Algorithm 2.

Lemma 4.2. Algorithm 2 is correct and runs in $O(n)$ time.

Proof. Let ρ be the root of the given binary cotree T . We will show that the coloring ϕ obtained by calling ASSIGNCOLORS($\rho, 1$) is an optimal acyclic (and star) coloring of G .

Recall that two distinct vertices in G are adjacent if and only if the lowest common ancestor of the corresponding leaves in T is a \otimes -node. It follows that any coloring that assigns disjoint sets of colors to the children of every \otimes -node will be a proper coloring. It is easy to see that both SATURATE() and ASSIGNCOLORS() satisfy this condition; hence ϕ is a proper coloring of G , which means that we may apply Lemma 3.2 to show that ϕ is an acyclic coloring of G . To this end, observe that the procedure SATURATE() behaves exactly as expected: the result of calling SATURATE(τ, K) is that the node τ is saturated by ϕ with the colors $\{K, \dots, |V_\tau| - 1\}$. Because SATURATE() is called on one of the two children of every \otimes -node, we see that ϕ satisfies condition (ii) of Lemma 3.2, from which it follows that ϕ is an acyclic coloring as desired.

It remains to be shown that ϕ uses exactly $\chi_a(G)$ colors. Our proof of this fact is based upon an inductive argument concerning the recursive calls ASSIGNCOLORS(τ, K). Trivially, when τ is a leaf, the induced subgraph G_τ is colored using $\chi_a(G_\tau) = 1$ colors. Now suppose τ is an internal node with children α and β . If τ is a \odot -node, and the calls ASSIGNCOLORS(α, K) and ASSIGNCOLORS(β, K) result in optimal acyclic colorings of G_α and G_β , then, by Lemma 2.1, the induced subgraph G_τ is colored using $\chi_a(G_\tau) = \max\{\chi_a(G_\alpha), \chi_a(G_\beta)\}$ colors. Now suppose τ is a \otimes -node and assume without loss of generality that $|V_\alpha| + \chi_a(G_\beta) \leq |V_\beta| + \chi_a(G_\alpha)$. If the call ASSIGNCOLORS($\beta, K + |V_\alpha|$) results in an

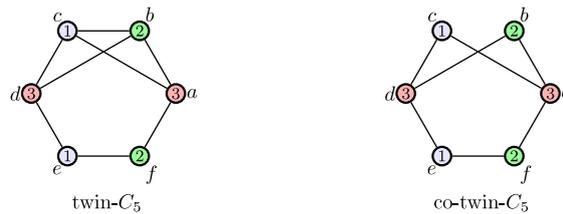


Fig. 4. Two graphs, each given with acyclic coloring ϕ such that neither can be ϕ -triangulated. In the graph on the left, we cannot add an edge incident on d or a without creating a bichromatic cycle or violating the condition that the coloring is proper. Therefore, the cycles induced by $\{b, d, e, f, a\}$ and $\{c, d, e, f, a\}$ must be triangulated by adding edges $\{b, e\}$ and $\{c, f\}$, respectively. This results in $\{c, b, e, f\}$ inducing a bichromatic cycle. Note that edge $\{c, b\}$ must be added in any triangulation of the graph on the right, which reduces the problem to that of the graph on the left.

optimal acyclic coloring of G_β , then, by Lemma 2.1, the induced subgraph G_τ is colored using $\chi_\alpha(G_\tau) = |V_\alpha| + \chi_\alpha(G_\beta)$ colors.

As in the proof of Lemma 4.1, we establish the running time via a one-to-one correspondence between recursive procedure calls and nodes in T . In this case, each node corresponds to either a call of ASSIGNCOLORS() or a call of SATURATE(). \square

5. Concluding remarks

Theorem 4 implies a natural heuristic for the acyclic coloring problem: simply find a triangulation G^+ of G that is close to optimal (with respect to treewidth), and then compute an optimal proper coloring of G^+ , using $O(n + m)$ time [17]. Here we use the fact that treewidth is a particularly well-studied parameter, and there are many heuristics, approximation algorithms, exact (exponential-time) algorithms, and polynomial-time algorithms for many classes of graphs [6,14,22]. In particular, for a constant k there is a linear-time algorithm for determining whether the treewidth of a graph is at most k and, if so, finding a corresponding triangulation [3].

Furthermore, Lemma 2.1 applies to any graph that is decomposable with respect to the join operation, and so it may be used as a reduction step that should be applied as the first step of any heuristic. Moreover, Lemma 2.1 implies that we can also find an optimal acyclic or star coloring of any graph for which these problems can be solved on all the graphs that result from recursively applying the join decomposition. For example, the *tree-cographs* [25] are those graphs that result by taking disjoint unions and joins of trees or other tree-cographs. The class of cographs is properly contained within this class. Since it is trivial to find an optimal acyclic or star coloring of a tree in linear time [13], it follows that we can solve these problems in linear time on the entire class of tree-cographs.

In the proof of Lemma 3.2, we were able to add at least one edge to every induced cycle on four vertices in G (which was given along with an acyclic coloring) such that no new induced cycles were created. However, one can easily construct an example for general graphs where this is not the case. Furthermore, there are graphs G with acyclic colorings ϕ for which G cannot be ϕ -triangulated. Two minimal examples are shown in Fig. 4.

In Lemma 3.2 we proved the equivalence of the acyclic coloring and treewidth problems for cographs by showing that every acyclic coloring of a cograph G is a proper coloring of some triangulation of G . It would be useful to prove similar results for other classes of graphs; it is natural to consider other classes for which the treewidth problem can be solved in polynomial time.

Acknowledgments

The author wishes to thank the anonymous referees, whose suggestions greatly improved the presentation of this paper. The majority of this work was conducted when the author was employed by The University of Chicago and Argonne National Laboratory, supported in part by US Department of Energy under Contract DE-AC02-06CH11357.

References

- [1] M.O. Albertson, G.G. Chappell, H.A. Kierstead, A. Kündgen, R. Ramamurthi, Coloring with no 2-colored P_4 s, Electron. J. Combin. 11 (1) (2004) http://www.combinatorics.org/Volume_11/Abstracts/v11i1r26.html.
- [2] H.L. Bodlaender, Achromatic number is NP-complete for cographs and interval graphs, Inform. Process. Lett. 31 (3) (1989) 135–138. doi:10.1016/0020-0190(89)90221-4.
- [3] H.L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317. doi:10.1137/S0097539793251219.
- [4] H.L. Bodlaender, M.R. Fellows, M.T. Hallett, H.T. Wareham, T.J. Warnow, The hardness of perfect phylogeny, feasible register assignment and other problems on thin colored graphs, Theoret. Comput. Sci. 244 (1–2) (2000) 167–188. doi:10.1016/S0304-3975(98)00342-9.
- [5] H.L. Bodlaender, R.H. Möhring, The pathwidth and treewidth of cographs, SIAM J. Discrete Math. 6 (2) (1993) 181–188. doi:10.1137/0406014.
- [6] H.L. Bodlaender, U. Rotics, Computing the treewidth and the minimum fill-in with the modular decomposition, Algorithmica 36 (4) (2003) 375–408. doi:10.1007/s00453-003-1026-5.
- [7] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

- [8] T.F. Coleman, J.-Y. Cai, The cyclic coloring problem and estimation of sparse Hessian matrices, *SIAM J. Algebr. Discrete Methods* 7 (1986) 221–235. doi:10.1137/0607026.
- [9] T.F. Coleman, J.J. Moré, Estimation of sparse Hessian matrices and graph coloring problems, *Math. Program.* 28 (3) (1984) 243–270. doi:10.1007/BF02612334.
- [10] D.G. Corneil, H. Lerchs, L.S. Burlingham, Complement reducible graphs, *Discrete Appl. Math.* 3(3) (1981) 163–174. doi:10.1016/0166-218X(81)90013-5.
- [11] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* 14 (4) (1985) 926–934. doi:10.1137/0214065.
- [12] G. Fertin, A. Raspaud, Acyclic coloring of graphs of maximum degree five: nine colors are enough, *Inform. Process. Lett.* 105 (2) (2008) 65–72. doi:10.1016/j.ipl.2007.08.022.
- [13] G. Fertin, A. Raspaud, B. Reed, Star coloring of graphs, *J. Graph Theory* 47 (3) (2004) 163–182. doi:10.1002/jgt.v47:3.
- [14] F.V. Fomin, D. Kratsch, I. Todinca, Y. Villanger, Exact algorithms for treewidth and minimum fill-in, *SIAM J. Comput.* 38 (3) (2008) 1058–1079. doi:10.1137/050643350.
- [15] A.H. Gebremedhin, F. Manne, A. Pothen, What color is your Jacobian? Graph coloring for computing derivatives, *SIAM Rev.* 47 (4) (2005) 629–705. doi:10.1137/S0036144504444711.
- [16] A.H. Gebremedhin, A. Tarafdar, F. Manne, A. Pothen, New acyclic and star coloring algorithms with application to computing Hessians, *SIAM J. Sci. Comput.* 29 (3) (2007) 1042–1072. doi:10.1137/050639879.
- [17] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, second ed., in: *Annals of Discrete Mathematics*, vol. 57, Elsevier, Amsterdam, The Netherlands, 2004.
- [18] B. Grünbaum, Acyclic colorings of planar graphs, *Israel J. Math.* 14 (4) (1973) 390–408. doi:10.1007/BF02764716.
- [19] M. Habib, C. Paul, A simple linear time algorithm for cograph recognition, *Discrete Appl. Math.* 145 (2) (2005) 183–197. doi:10.1016/j.dam.2004.01.011.
- [20] K. Jansen, P. Scheffler, Generalized coloring for tree-like graphs, *Discrete Appl. Math.* 75 (2) (1997) 135–155. doi:10.1016/S0166-218X(96)00085-6.
- [21] S.K. Kannan, T.J. Warnow, Triangulating 3-colored graphs, *SIAM J. Discrete Math.* 5 (2) (1992) 249–258. doi:10.1137/0405019.
- [22] T. Kloks, *Treewidth: Computations and Approximations*, in: *Lecture Notes in Computer Science*, vol. 842, Springer, 1994, doi:10.1007/BFb0045375.
- [23] A. Lyons, Acyclic and star colorings of joins of graphs and an algorithm for cographs, in: *CTW, 2009*, pp. 199–202. <http://www.lix.polytechnique.fr/ctw09/ctw09-proceedings.pdf#page=211>.
- [24] S. Skulrattanakulchai, Acyclic colorings of subcubic graphs, *Inform. Process. Lett.* 92 (4) (2004) 161–167. doi:10.1016/j.ipl.2004.08.002.
- [25] G. Tinhofer, Strong tree-cographs are Birkhoff graphs, *Discrete Appl. Math.* 22 (3) (1989) 275–288. doi:10.1016/0166-218X(88)90100-X.