Information Processing Letters 24 (1987) 133–137
North-Holland

30 January 1987

THE MAXIMUM k-COLORABLE SUBGRAPH PROBLEM FOR CHORDAL GRAPHS

Mihalis YANNAKAKIS

AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, U.S.A.

Fanica GAVRIL

Computer Science Division, University of California, Davis, CA 95616, U.S.A.

Communicated by M.A. Harrison
Received 18 December 1985

We discuss the problems of finding maximum and connected maximum k-colorable subgraphs in chordal graphs. We prove that the problems are polynomially solvable when k is fixed and NP-hard when k is not fixed. As a special case, we can find in polynomial time the maximum induced tree and forest of a chordal graph.

Keywords: Chordal graph, k-colorable

1. Introduction

We consider finite undirected graphs G(V, E), with no parallel edges and no self-loops, where V is the set of the graph nodes and E is the set of its edges. For a subset U of V, the *induced subgraph* G(U) of G is the graph whose set of nodes is U, two nodes being adjacent in G(U) if and only if they are adjacent in G. A *clique* is a maximal set of nodes having every two elements adjacent. A *maximum clique* is a clique with a maximum number of elements. A set of nodes is called *independent* if no two of its elements are adjacent. A subgraph G(U) of G is said to be k-colorable if its nodes can be colored with k colors such that every two adjacent nodes have different colors. The chromatic number of a graph G(V, E) is the minimum k such that G(V, E) is k-colorable. A graph G is *perfect* if all of its subgraphs have chromatic number equal to the size of a maximum clique.

A graph G(V, E) is called *chordal* if every simple cycle with more than three nodes has an edge connecting two nonconsecutive nodes. A survey of results and applications about these graphs can be found in [5].

In the present paper we shall discuss the following problem: given a graph G(V, E) and a positive integer k, find a maximum k-colorable subgraph G(U); equivalently, find k independent sets covering a maximum number of nodes. We shall call this problem the *maximum k-colorable subgraph problem*. When we request that G(U) is connected, we call it the *connected maximum k-colorable subgraph problem*.

Coloring the maximum number of nodes with k colors is NP-complete in general since it includes as special cases both the maximum independent set problem ($k = 1$) and the chromatic number problem (see [3] for a survey on NP-complete problems).

The complexity of the maximum k-colorable subgraph problem becomes interesting on perfect graphs [9], because for such graphs one can solve in polynomial time both special cases, as well as the complementary problems of finding a maximum clique and covering the nodes with the minimum number of cliques [8]. In fact, for comparability graphs and their complements (one 'classical' class of perfect graphs) the k-coloring problem can be solved in polynomial time: Greene [6] and Greene and Kleitman [7] prove an elegant min–max relation, and Frank [2] presents a polynomial-time algorithm.

In the present paper we show that, for chordal graphs and their complements (the other classical class of perfect graphs), the problem is NP-complete. When k is fixed, we present polynomial-time algorithms to solve the maximum and the connected maximum k-colorable subgraph problems for chordal graphs. In the special case of interval graphs we show that a simple greedy algorithm finds a maximum k-colorable subgraph even when k is not fixed.

A bipartite subgraph of a chordal graph is in fact a forest since a chordal graph has no chordless cycles and a bipartite graph has no triangles. Therefore, in a chordal graph, the maximum 2-colorable subgraph problem is equivalent to finding the maximum induced forest, and the connected maximum 2-colorable subgraph problem is equivalent to finding the maximum induced tree. Hence, we obtain polynomial-time algorithms for these problems. In comparison, on bipartite graphs these problems are NP-complete: The NP-completeness of the maximum forest problem on bipartite graphs was shown in [11], and that of the maximum tree problem can easily be derived.

2. Chordal graphs

Consider a graph G(V, E). We shall denote by C its set of cliques and by C_v the set of cliques containing a node v. Let $\bar{C} = \{\bar{c} \mid c \in C\}$ be a set of nodes, and for every $v \in V$ let $\bar{C}_v = \{\bar{c} \mid c \in C_v\}$. As proven in [4], a graph G is chordal if and only if there exists a tree T whose node set is $\bar{C}$ such that, for every node v of G, the subgraph $T_v = T(\bar{C}_v)$ of T induced by $\bar{C}_v$ is a subtree (i.e., is connected). That is, a graph is chordal if and only if it is the intersection graph of a family of subtrees in a tree. The tree T and the family of subtrees $\{T_v \mid v \in V\}$ is called a *tree representation* of G. Using the recognition algorithm for chordal graphs described in [10] and the fact that a chordal graph G(V, E) has at most $|V|$ cliques (see [5]), the algorithm described in [4] for constructing a tree representation for a chordal graph can be implemented in $O(|E|)$ steps. Before going further, we prove the following theorem.

Theorem 2.1. *Consider a perfect graph* G(V, E). *A subgraph* G(U) *of* G *is* k*-colorable if and only if it satisfies* $|c \cap U| \leqslant k$ *for every clique* c *of* G.

Proof. The one direction is obvious: If G(U) is a k-colorable subgraph of G, then for every clique c of G we have $|c \cap U| \leqslant k$. Conversely, suppose that G(U) satisfies $|c \cap U| \leqslant k$ for every clique c of G. Since every clique of G(U) is equal to $c \cap U$ for some clique c of G, it follows that the maximum clique of G(U) has at most k nodes. Since G is perfect, this implies that G(U) is k-colorable. □

We shall now describe an algorithm for finding a maximum k-colorable subgraph of a chordal graph G(V, E). Let T and $\{T_v \mid v \in V\}$ be tree representations of G. We consider T a rooted tree by making one of its nodes the root. For every node $\bar{c}$ of T let $T_{\bar{c}}$ be the subtree of T rooted at $\bar{c}$ and let $V_{\bar{c}} = \bigcup \{d \mid \bar{d}$ is a node of $T_{\bar{c}}\}$. We shall perform a dynamic programming computation starting with the leaves of T and working toward the root. For every node $\bar{c}$ of T and every subset c′ of c with at most k elements we shall compute a maximum k-colorable subgraph H(c, c′) of $G(V_{\bar{c}})$ satisfying the additional restriction that, among the nodes of c, exactly the elements of c′ are colored; denote by b(c, c′) the number of nodes in H(c, c′).

Assume that we have already found these subgraphs for all the sons $\bar{c}_1, \ldots, \bar{c}_r$ of a node $\bar{c}$ of T. Consider a k-colorable subgraph of $G(V_{\bar{c}})$ which contains the subset c′ of c, and let N be its set of nodes. Let $c_i' = N \cap c_i$ be the set of nodes in c_i that are colored, and let $N_i = N \cap V_{\bar{c}_i}$ be the set of colored nodes in the subtree hanging from $\bar{c}_i$; note that $c_i' = N_i \cap c_i$. We observe that the sets $V_{\bar{c}_1}, \ldots, V_{\bar{c}_r}$ (as well as $N_1, \ldots, N_r$) intersect only at nodes of c. Futhermore,

$$N \cap c \cap c_i = c_i' \cap c = c' \cap c_i.$$

Thus, N can be written as $c' \cup (\bigcup_{i=1}^{r} N_i)$, where, for each $i = 1, 2, \ldots, r$, the set N_i induces a k-colorable subgraph of $G(V_{\bar{c}_i})$ and its intersection c_i' with c_i satisfies the equation

$$c_i' \cap c = c' \cap c_i. \tag{1}$$

Conversely, suppose that, for each $i = 1, 2, \ldots, r$, we have a set N_i which induces a k-colorable subgraph of $G(V_{\bar{c}_i})$, and whose intersection c_i' with c_i satisfies (1). Let N be the union of c' and the N_i's. From the definition of tree representation, if an element is in both N_i and c, then it has to be in c_i also. Therefore,

$$N_i \cap c = N_i \cap c_i \cap c = c_i' \cap c \subseteq c'.$$

Thus, $N \cap c = c'$. Similarly, if an element of N_i is in $V_{\bar{c}_j}$ with $i \neq j$, then it has to be in c and c_j also; therefore,

$$N_i \cap V_{\bar{c}_j} \subseteq N_i \cap c \cap c_j \subseteq c' \cap c_j \subseteq c_j'.$$

This implies in particular that, for any node $\bar{d}$ of $T_{\bar{c}}$ other than $\bar{c}$, say a node in the *ith* subtree, the intersection $N \cap d$ is equal to $N_i \cap d$, and therefore has at most k elements. It follows from Theorem 2.1 that N induces a k-colorable subgraph which contains exactly the subset c' of c. The size of N is

$$|N| = \sum_{i=1}^{r} |N_i - c'| + |c'|$$
$$= \sum_{i=1}^{r} \left(|N_i| - |c' \cap c_i| \right) + |c'|,$$

since the N_i's intersect only at elements of c'.

It follows from our discussion that $b(c, c')$ can be computed by the expression

$$b(c, c') = \sum_{i=1}^{r} \left(\max_{c_i'} b(c_i, c_i') - |c' \cap c_i| \right) + |c'|, \tag{2}$$

where c_i' ranges over all subsets of c_i with at most k elements, satisfying $c' \cap c_i = c_i' \cap c$. The subgraph $H(c, c')$ is obtained by taking the union of c' and the subgraphs $H(c_i, c_i')$ where c_i' is the subset of c_i on which the maximum has been obtained in (2). When we have these maximum k-colorable subgraphs $H(c, c')$ for the root $\bar{c}$ of T, a maximum one among them will be a maximum k-colorable subgraph of G.

The above algorithm also works for the weighted version of the maximum k-colorable subgraph problem where the nodes have weights, and we want to maximize the sum of the weights of the colored nodes. Just replace $|c' \cap c_i|$ and $|c'|$ in (2) with the sum of the weights of the elements of $c' \cap c_i$ and c' respectively.

To obtain an algorithm for the connected maximum k-colorable subgraph problem we request in addition that expression (2) and the corresponding subgraphs be computed only on the c_i's satisfying $c' \cap c_i \neq \emptyset$. Therefore, we state the following theorem.

Theorem 2.2. *The maximum weight and the connected maximum weight* k*-colorable subgraph problems in chordal graphs are polynomially solvable when* k *is fixed.*

Corollary 2.3. *The problems of finding maximum tree and forest subgraphs in a chordal graph are polynomially solvable.*

The complementary problem of covering the maximum number of nodes with k cliques, when k is fixed, can trivially be solved in polynomial time on chordal graphs, since a chordal graph has at most $|V|$ cliques. We shall now show that both problems become NP-complete when k is not fixed. We shall use the same reduction for both problems.

A *split graph* is a graph whose nodes can be partitioned into two subsets I and K such that I is an independent set and K induces a complete graph. Two elementary properties of these graphs are: (i) The complement of a split graph is also a split graph, and (ii) split graphs are chordal (see [5,9] for more information).

Theorem 2.4. *When* k *is not fixed, the maximum* k*-colorable subgraph problem for split graphs (and, thus, also chordal graphs and their complements) is* NP*-complete.*

Proof. We reduce the Set Covering problem to the problem of covering the maximum number of nodes of a split graph with k cliques. In the Set Covering problem we are given a set X =

$\{x_1, \ldots, x_n\}$, a family $\mathcal{F}$ of subsets of X, and an integer k. The problem is to determine whether we can cover all elements of X using at most k sets from $\mathcal{F}$.

Given an instance X, $\mathcal{F}$, k of the Set Covering problem, construct a split graph G with nodes $X \cup \mathcal{F}$ as follows. The set X induces a clique in G, $\mathcal{F}$ is an independent set, and in addition we include an edge between an element s of $\mathcal{F}$ and an element x_i of X iff $x_i \in s$. We claim that there is a solution to the Set Covering problem if and only if we can cover at least $n + k$ nodes in G using k cliques.

First note that G has the following cliques: X and, for each $s \in \mathcal{F}$, a clique consisting of node s and the members of s. Suppose that there are k sets in $\mathcal{F}$ which cover X; the cliques of G corresponding to these k sets cover $n + k$ nodes (X and k nodes in $\mathcal{F}$). Conversely, suppose that we can find k cliques of G which cover $n + k$ nodes. Since X has n nodes and every clique of G contains at most one node in $\mathcal{F}$, it follows that each of the k cliques contains exactly one node of $\mathcal{F}$, and the corresponding sets cover X. □

3. Interval graphs

When subfamilies of chordal graphs have additional properties, the maximum k-colorable subgraph problem may become polynomial even when k is not fixed. For example, the problem is polynomially solvable for interval graphs, using the algorithm in [2], since their complements are comparability graphs. Moreover, there exists a simple efficient greedy algorithm solving this problem for interval graphs: The set of intervals representing an interval graph G(V, E) is processed from left to right in increasing order of the right endpoints. For a node v let $\bar{v}$ denote its corresponding interval. Having a maximum k-colorable subgraph G(U′) for the set of nodes already processed, the next node v is added to U′ if $G(U' \cup \{v\})$ contains no clique with more than k nodes, and is discarded otherwise. A set of intervals representing the interval graph can be constructed in $O(|V| + |E|)$ time using the algorithm of Booth and Lueker [1], and the greedy algorithm can easily be implemented within the same time bound. Let G(U) be the subgraph obtained by this process and let G(W) be a maximum k-colorable subgraph of G such that $|W \cap U|$ is maximum. We claim that $U = W$.

Assume for the sake of contradiction that $W \neq U$. Index the nodes of U in increasing order of the right endpoints of their intervals. Let i be the smallest index such that the *ith* element v_i of U is not in W. Let r be the right endpoint of the corresponding interval $\bar{v}_i$. From the definition of i, all intervals of U which finish (i.e., have their right endpoint) before r are also in W. Since no interval is rejected from U unless it forms a clique of size k with intervals that were already selected, all intervals in $W - U$ finish after r. Let w be the element of $W - U$ whose corresponding interval has the leftmost left endpoint; let ℓ be the left endpoint of $\bar{w}$. Let W′ be the set obtained from W by replacing w by v_i. We claim that W′ is also k-colorable; this would contradict our choice of W.

Recall that a clique of an interval graph corresponds in the interval model to the set of intervals which cross over some point of the real line. Let p be such a point. If $p < \ell$, then any interval of W′ crossing over p has its left endpoint before ℓ, and therefore is also in U. If $p > r$, then any interval of W′ crossing over p is also in W. Finally, if $\ell \leqslant p \leqslant r$, then among the intervals that cross p we have removed one from W, namely w, and added (at most) another one, v_i. In any case, since U and W are k-colorable, at most k intervals of W′ cross any point, and therefore W′ is also k-colorable.

The weighted version of the maximum k-colorable subgraph problem for interval graphs is also polynomially solvable since we can write it as an integer LP problem on the nodes vs. cliques matrix which is totally unimodular.

References

[1] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (1976) 335–379.

[2] A. Frank, On chain and antichain families of a partially ordered set, J. Combin. Theory Ser. B 29 (1980) 176–184.

[3] M.R. Garey and D.S. Johnson, Computer and Intractabil-

ity: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, CA, 1978).

[4] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Combin. Theory Ser. B 16 (1974) 47–56.

[5] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs (Academic Press, New York, 1980).

[6] C. Greene, Some partitions associated with a partially ordered set, J. Combin. Theory Ser. A 20 (1976) 69–79.

[7] C. Greene and D. Kleitman, The structure of Sperner k-families. J. Combin. Theory Ser. A 20 (1976) 41–68.

[8] M. Grötschel, L. Lovasz and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981) 169–197.

[9] L. Lovasz, Perfect graphs, in: L.W. Beineke and R.J. Wilson, eds., Selected Topics in Graph Theory, Vol. 2 (Academic Press, New York, 1983) 55–87.

[10] D.J. Rose, R.E. Tarjan and G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput. 5 (1976) 266–283.

[11] M. Yannakakis, Node-deletion problems on bipartite graphs, SIAM J. Comput. 10 (1981) 310–327.