ELSEVIER

Discrete Applied Mathematics 66 (1996) 189–203

DISCRETE APPLIED MATHEMATICS

Algorithmic aspects of the generalized clique-transversal problem on chordal graphs

Maw-Shang Chang[a,1], Yi-Hua Chen[a,1], Gerard J. Chang[b,*,2,3], Jing-Ho Yan[b,2]

[a] *Department of Computer Science and Information Engineering, National Chung Cheng University, Ming-Shiun, Chiayi 621, Taiwan*

[b] *Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan*

Received 11 March 1993; revised 29 November 1994

Abstract

Suppose $G=(V,E)$ is a graph in which each maximal clique C_i is associated with an integer r_i, where $0 \leqslant r_i \leqslant |C_i|$. The generalized clique transversal problem is to determine the minimum cardinality of a subset D of V such that $|D \cap C_i| \geqslant r_i$ for every maximal clique C_i of G. The problem includes the clique-transversal problem, the $i,1$ clique-cover problem, and for perfect graphs, the maximum q-colorable subgraph problems as special cases. This paper gives complexity results for the problem on subclasses of chordal graphs, e.g., strongly chordal graphs, k-trees, split graphs, and undirected path graphs.

Keywords: Clique-transversal set; Neighborhood number; Domination; Dual; Chordal graph; Strongly chordal graph; k-Tree; Split graph; Undirected path graph

1. Introduction

In this paper, $G = (V,E)$ represents a graph with vertex set V of size n and edge set E of size m. A *clique* is a subset of pairwise adjacent vertices of V. An *i-clique* is a clique of size i. A *maximal clique* is a clique that is not a proper subset of any other clique. Denote by $\mathscr{C}(G)$ the set of all maximal cliques of G.

This paper studies the following generalized clique-transversal problem, which includes many clique-related problems as special cases. Suppose each maximal clique C_i of G is associated with an integer r_i, where $0 \leqslant r_i \leqslant |C_i|$. Let $R = \{(C_i, r_i) : C_i \in \mathscr{C}(G)\}$. An *R-clique-transversal set* of G is a subset D of V such that $|D \cap C_i| \geqslant r_i$ for every $C_i \in \mathscr{C}(G)$. The *generalized clique-transversal problem* is, given a graph

[*] Corresponding author. E-mail: gjchang@math.nctu.edu.tw.

[1] Supported partly by the National Science Council under grant NSC81-0408-E194-04.

[2] Supported partly by the National Science Council under grant NSC82-0208-M009-050.

[3] Supported partly by DIMACS, Rutgers University.

SSDI 0166-218X(95)00048-8

G and R, to determine the *R-clique-transversal number* $\tau_R(G)$ that is the minimum size of an R-clique-transversal set of G. The problem has a natural dual, as follows. A *clique-independent set* is a collection of pairwise disjoint maximal cliques. The *R-clique-independence problem* is, given a graph G and R, to determine the *R-clique-independence number* $\alpha_R(G)$ of G that is the maximum value $\sum_{C_i \in \mathscr{P}} r_i$ for a clique-independent set $\mathscr{P}$ of G. For any R-clique-transversal set D and any clique-independent set $\mathscr{P}$,

$$\sum_{C_i \in \mathscr{P}} r_i \leqslant \sum_{C_i \in \mathscr{P}} |D \cap C_i| = |D \cap (\bigcup_{C_i \in \mathscr{P}} C_i)| \leqslant |D \cap V| = |D|.$$

Hence, we have the following *weak duality inequality*: for any graph G,

$$\alpha_R(G) \leqslant \tau_R(G). \tag{1}$$

Note that the inequality may be strict, as $\alpha_R(C_{2n+1}) = n < n+1 = \tau_R(C_{2n+1})$ when all $r_i = 1$ and $n \geqslant 2$.

The concept of chordal graphs was introduced by Hajnal and Surányi [24] in connection with the theory of perfect graphs (see [21]). A graph is *chordal* (or *triangulated*) if every cycle of length greater than three has a chord (i.e., every induced cycle is a triangle). The *neighborhood* $N(v)$ of a vertex v is the set of all vertices adjacent to v. The *closed neighborhood* $N[v]$ of v is $\{v\} \cup N(v)$. One of the most important properties of a chordal graph $G = (V,E)$ is that its vertices have a *perfect elimination ordering*, i.e., an ordering $v_1, v_2, \ldots, v_n$ of V such that $N_i[v_i]$ is a clique for $1 \leqslant i \leqslant n$, where $N_i[v_j] = \{v_k \in N[v_j] : i \leqslant k\}$. Note that any maximal clique of a chordal graph G is equal to some $N_i[v_i]$, but an $N_i[v_i]$ is not necessarily a maximal clique. Consequently, a chordal graph has at most n maximal cliques. It is also known that all maximal cliques can be enumerated in $O(m+n)$ time (see [14, 16]). In this paper, we study algorithmic aspects of the generalized clique-transversal problem and related problems on subclasses of chordal graphs, such as strongly chordal graphs, k-trees, split graphs, and undirected path graphs.

If we restrict $r_i = 1$ for all maximal cliques C_i of G, the generalized clique-transversal (resp. R-clique-independence) problem becomes the *clique-transversal* (resp. *clique-independence*) *problem.* The corresponding *clique-transversal* (resp. *clique-independence*) *number* is denoted by $\tau_C(G)$ (resp. $\alpha_C(G)$). The above weak duality inequality becomes: for any graph G,

$$\alpha_C(G) \leqslant \tau_C(G). \tag{2}$$

$\alpha_C(G)$ and $\tau_C(G)$ have been studied in [1, 5, 10, 36]. In particular, it was proven that determining $\alpha_C(G)$ and $\tau_C(G)$ for a split graph G is NP-complete and that there are linear algorithms for finding $\alpha_C(G)$ and $\tau_C(G)$ of a strongly chordal graph G (see [5]).

Another special case of the generalized clique-transversal problem is the $C_{i,1}$ problem described as follows. This problem has in fact been studied in a more general setting (see [8]). For $i \geqslant j \geqslant 1$, the *$C_{i,j}$ problem* is to determine the *i,j clique cover number*

$c_{i,j}(G)$ of a graph G, which is the minimum number of j-cliques such that every i-clique of G includes such a j-clique. It is not hard to see that the $C_{i,1}$ problem is the same as the generalized clique-transversal problem with $r_j = \max\{|C_j| - i + 1, 0\}$ for each maximal clique C_j of G. A classical result of the $C_{i,j}$ problem is Turán's theorem, which states that $c_{i,2}(K_n)$ is equal to the number of edges in a complete $(i-1)$-partite graph of n vertices, where any two distinct partite sets have sizes that differ by at most one. The problem has also been studied from an algorithmic point of view by many authors. In particular, it was shown that it is NP-complete on general graphs when $i > j \geqslant 1$ [7, 15, 28, 37], on chordal graphs when $i > j \geqslant 2$ [7], on split graphs when $i - 1 = j \geqslant 2$ [7], and on split graphs when $i \geqslant 3 > 1 = j$ and i is part of the input [8]. On the other hand, there is a polynomial time algorithm for the $C_{2,1}$ problem, which is exactly the vertex cover problem and is reducible to the maximum independent set problem, on chordal graphs [16], a polynomial-time algorithm for the $C_{i,1}$ problem on chordal graphs when i is fixed, and an $O(m+n)$ time algorithm for the $C_{i,1}$ problem on interval graphs even when i is not fixed [31].

The *maximum q-colorable subgraph problem* is, given a graph $G = (V, E)$, to determine the maximum size $s_q(G)$ of a subset of V that can be partitioned into q disjoint independent sets. The problem has been studied by [13, 20, 22, 23, 38]. For any fixed q, the problem is NP-complete for general graphs. It was shown in [8] that for any perfect graph G, $s_q(G) = |V| - c_{q+1,1}(G)$. Consequently, the maximum q-colorable subgraph problem is equivalent to the $C_{q+1,1}$ problem for chordal graphs.

The rest of this paper is organized as follows. Section 2 reviews some terminology that will be applied in this paper. Section 3 gives an $O(m+n)$ time algorithm for the generalized clique-transversal problem on strongly chordal graphs provided that strong elimination orderings are known in advance. We modify this algorithm to get an algorithm for the $C_{i,1}$ problem so that we do not need to find all maximal cliques explicitly. Section 4 gives a linear time algorithm for the generalized clique-transversal problem on k-trees when k is fixed. Section 5 discusses the NP-complete results on k-trees (with unbounded k) and undirected path graphs.

2. Preliminaries

In this paper, all graphs are finite, undirected, and without loops or parallel edges. A graph is an *intersection graph* if there is a correspondence between its vertices and a family of sets (the *intersection model*) such that two vertices are adjacent in the graph if and only if their two corresponding sets have a nonempty intersection. Restricting the sets to subtrees of a tree determines the class of chordal graphs [17]. If the intersection models are further restricted so that each subtree is a path, a proper subclass called the *undirected path graphs* results [19]. Further restricting the model to rooted trees with paths directed away from the root yields the *directed path graphs* [18]. Requiring that the tree itself be a path defines the class of *interval graphs* [14, 29].

As Gavril has shown in his papers, the intersection models for chordal graphs can always be chosen so that the nodes of the tree are the maximal cliques of the original graph. Each vertex of the graph then corresponds to the subtree comprising of exactly those maximal cliques to which it belongs. We call such an intersection model a *clique tree* for the graph.

Other two related subclasses of chordal graphs are split graphs and k-trees. A *split graph* is a graph whose vertex set can be partitioned into the disjoint union of an independent set and a clique. This concept was first introduced by Földes and Hammer [12], who also proved that a graph is split if it and its complement are chordal. A k-tree is defined recursively as follows: a K_k is a k-tree, and if G is a k-tree then so is the graph formed by adding a new vertex to G and making it adjacent to all vertices of a k-clique in G. In a k-tree of more than k vertices, each maximal clique is of size $k+1$, which is formed by joining a new vertex to a previous existing k-clique. A clique tree of a k-tree thus can be obtained recursively as follows: K_1 is a clique tree of K_{k+1}, and if T is a clique tree of a k-tree G and G' is obtain from G by adding a new vertex v' adjacent to all vertices in a k-clique C of G, then a clique tree T' of G' can be formed by adding a new vertex v' adjacent only to v in T, where v corresponds to a $(k+1)$-clique of G that includes C.

In conjunction with the study of domination in graph theory, the following subclass of chordal graphs was studied in [6, 11, 26]. A *strongly* (or *sun-free*) *chordal graph* is a graph $G = (V, E)$ whose vertex set has a *strong elimination ordering*, i.e., an ordering $v_1, v_2, \ldots, v_n$ of V such that $i \leqslant j \leqslant k$ and $v_j, v_k \in N_i[v_i]$ imply $N_i[v_j] \subseteq N_i[v_k]$. Note that a strong elimination ordering is a perfect elimination ordering. To date, the fastest algorithm to recognize a strongly chordal graph and give a strong elimination ordering takes $O(m \log n)$ (see [32]) or $O(n^2)$ time (see [35]). Strongly chordal graphs include trees, block graphs, powers of trees, interval graphs, and directed path graphs.

A *hypergraph* is an ordered pair $H = (V, E)$ consisting of a finite nonempty set V of *vertices* and a collection E of nonempty subsets of V called *(hyper)edges*. A hypergraph is *k-uniform* if each edge is of size k. A 2-uniform hypergraph is just a graph. The *transversal number* $\tau(H)$ of a hypergraph H is the minimum size of a subset of vertices that meets all edges of H. The *matching number* $m(H)$ of H is the maximum size of a pairwise disjoint subclass of E. For more terminology on hypergraphs see [2].

3. Algorithm on strongly chordal graphs

In this section we give a linear time algorithm for solving the generalized clique-transversal problem on strongly chordal graphs. Suppose G is a strongly chordal graph in which every maximal clique C_i is associated with an integer r_i, where $0 \leqslant r_i \leqslant |C_i|$. Let $R = \{(C_i, r_i) : C_i \in \mathscr{C}(G)\}$. We assume that a strong elimination ordering $v_1, v_2, \ldots, v_n$ is given. Note that a maximal clique of G is equal to some $N_i[v_i]$, but an $N_i[v_i]$ is not necessarily a maximal clique. For simplicity, we may assume that $N_i[v_i]$ is associated with a number $r_i = 0$ when it is not a maximal clique. It is not hard to see that D is an R-clique-transversal set of G if and only if $|N_i[v_i] \cap D| \geqslant r_i$ for all $1 \leqslant i \leqslant n$.

The algorithm is a greedy one. Initially, $D = V$ is an R-clique transversal set of G. It processes the vertices in the order $v_1, v_2, \ldots, v_n$. At iteration i, v_i is removed from D if $|N_j[v_j] \cap D| > r_j$ for all vertices $v_j \in S_i$, where

$$S_i = \{v_j \in N[v_i] : j \leqslant i\}.$$

Note that the new D is still an R-clique transversal set of G. At the completion of the algorithm, D is a minimum R-clique-transversal set of G.

Algorithm GCT. Solve the generalized clique-transversal problem on strongly chordal graphs.

Input. A strongly chordal graph G with a strong elimination ordering $v_1, v_2, \ldots, v_n$. Each maximal clique $C_i = N_i[v_i]$ is associated with an integer r_i with $0 \leqslant r_i \leqslant |C_i|$ and each nonmaximal clique $N_i[v_i]$ is associated with $r_i = 0$.

Output. A minimum R-clique-transversal D of G.

Method.

```
$D := V$;
for $i = 1$ to $n$ do
        if $|N_j[v_j] \cap D| > r_j$ for every vertex $v_j$ in $S_i$
        then $D := D - \{v_i\}$;
        end for.
```

Theorem 1. *Algorithm GCT solves the generalized clique-transversal problem on strongly chordal graphs in* $O(m + n)$ *time.*

Proof. Let D^* denote the final D when the algorithm stops. We shall prove that D^* is a minimum R-clique-transversal set of G. Since $|N_j[v_j] \cap D| \geqslant r_j$ for all $1 \leqslant j \leqslant n$ at any iteration, D^* is an R-clique-transversal set of G.

Choose a minimum R-clique-transversal set M such that $|M \cap D^*|$ is maximum. We claim that $D^* = M$. Suppose $D^* \neq M$. Then $D^* - M \neq \emptyset$, for otherwise the R-clique-transversal set D^* is a proper subset of M, which contradicts the minimality of M. Choose a vertex $v_p \in D^* - M$. Suppose $M - D^* = \emptyset$, i.e., M is a subset of D^*. At iteration p, for all $v_j \in S_p$,

$$|N_j[v_j] \cap D| \geqslant |N_j[v_j] \cap (M \cup \{v_p\})| \geqslant |N_j[v_j] \cap M| + 1 > r_j.$$

Then v_p must be removed from D at iteration p, a contradiction. Therefore, $M - D^* \neq \emptyset$. Let h be the smallest index such that $v_h \in M - D^*$ and

$$Q = \{v_j : |N_j[v_j] \cap (M - \{v_h\})| < r_j\}.$$

By the assumption that M is a minimum R-clique-transversal set, $Q \neq \emptyset$. It is also the case that $Q \subseteq S_h \subseteq N[v_h]$. Since $|N_j[v_j] \cap D^*| \geqslant r_j$, by the definition of Q, $N_j[v_j] \cap (D^* - M) \neq \emptyset$ for all vertices v_j of Q. Let s be the smallest index such that $v_s \in Q$ and t be the smallest index such that $v_t \in N_s[v_s] \cap (D^* - M)$. By the minimality of s and the fact that $Q \subseteq N[v_h]$, $Q \subseteq N_s[v_h]$.

For the case where $t < h$, by the minimality of h and the algorithm, at the beginning of iteration t, $M \cup \{v_t\} \subseteq D$ and so $|N_j[v_j] \cap D| \geqslant |N_j[v_j] \cap (M \cup \{v_t\})| > r_j$ for all vertices v_j of S_t. Therefore v_t must be removed from D by the algorithm; this contradicts the assumption that $v_t \in D^*$.

For the case where $t \geqslant h$, we have $s \leqslant h \leqslant t$. By the definition of a strong elimination ordering, $N_s[v_h] \subseteq N_s[v_t]$. Thus, $Q \subseteq N_s[v_h] \subseteq N_s[v_t]$. Let $M' = (M - \{v_h\}) \cup \{v_t\}$. If $v_j \notin Q$, then $|N_j[v_j] \cap M'| \geqslant |N_j[v_j] \cap (M - \{v_h\})| \geqslant r_j$ by the definition of Q. If $v_j \in Q$, then $j \leqslant h \leqslant t$, $v_jv_t \in E$, and $|N_j[v_j] \cap M'| \geqslant |N_j[v_j] \cap M| - 1 + 1 \geqslant r_j$. Thus, M' is a minimum R-clique-transversal set of G with $|M' \cap D^*| > |M \cap D^*|$, a contradiction to the maximality of $|M \cap D^*|$. So, $D^* = M$ is a minimum R-clique-transversal set of G.

To implement the algorithm efficiently, we associate each vertex v_j with a variable $d(v_j)$, which is equal to $|N_j[v_j] \cap D|$. So we can check the condition $|N_j[v_j] \cap D| > r_j$ in a constant time. Initially, $d(v_j) = |N_j[v_j]|$ for $1 \leqslant j \leqslant n$. If v_i is removed from D at iteration i, then $d(v_j)$ is decreased by 1 for all vertices v_j in S_i. Altogether, iteration i costs $O(d_i)$ time, where d_i is the degree of v_i in G. Hence the running time of the algorithm is $O(\sum_{i=1}^{n}(d_i + 1)) = O(m + n)$. □

Corollary 2. *The clique-transversal problem, the $C_{i,1}$ problem, and the maximum q-colorable subgraph problem can be solved in $O(m + n)$ time on strongly chordal graphs.*

To solve the problems in the above corollary by Algorithm GCT, we first need to identify all maximal cliques of G. Although this can be done in a linear time, we shall attempt to avoid this. Note that the algorithm in [5] for the clique-transversal problem on a strongly chordal graph G is able to avoid finding all maximal cliques. The algorithm chooses $N_i[v_i]$ in a special way to guarantee that the final output does get maximal cliques. For the $C_{i,1}$ problem, and so the q-colorable subgraph problem, the following modification of Algorithm GCT serves this purpose.

Algorithm Ci1. Solve the $C_{i,1}$ problem on strongly chordal graphs.

```
Input. A strongly chordal graph $G$ with a strong elimination ordering $v_1, v_2, \ldots, v_n$.
Output. A minimum $i,1$ clique cover $D$ of $G$.
Method.
    $D := \emptyset$;
    for $k = 1$ to $n$ do $s_k := 0$;
    for $j = 1$ to $n$ do
            if $s_k < i - 1$ for every vertex $v_k$ in $S_j$
            then $s_k := s_k + 1$ for every vertex $v_k$ in $S_j$
            else $D := D \cup \{v_j\}$;
            end for.
```

Note that at any iteration j, every s_k is simply the number of processed vertices in $N_k[v_k] - D$. The algorithm always keeps this value less than i to ensure that D remains an $i,1$ clique cover. The correctness of Algorithm Ci1 can be verified by either a similar argument as in Theorem 1 or a transformation as follows. Note that

$$\begin{aligned} & D \text{ is an } i,1 \text{ clique of } G \\ \Longleftrightarrow\ & |N_k[v_k] - D| \leqslant i-1 \quad \text{for } 1 \leqslant k \leqslant n \\ \Longleftrightarrow\ & |N_k[v_k] \cap D| \geqslant |N_k[v_k]| - i + 1 \quad \text{for } 1 \leqslant k \leqslant n \\ \Longleftrightarrow\ & |N_k[v_k] \cap D| \geqslant \max\{|N_k[v_k]| - i + 1, 0\} \quad \text{for all } N_k[v_k] \in \mathscr{C}(G). \end{aligned}$$

So the condition in the "if" statement of Algorithm Ci1 is the same as that in Algorithm GCT. We can also modify the algorithm to get one for the maximum q-colorable subgraph problem, which is equivalent to the $C_{q+1,1}$ problem.

4. Algorithm on k-trees with bounded k

This section establishes a linear time algorithm for the generalized clique-transversal problem on k-trees when k is bounded. Suppose $G = (V,E)$ is a k-tree in which each maximal clique C_i is associated with an integer r_i, where $0 \leqslant r_i \leqslant |C_i|$. Let T be a clique tree of G, which is considered to be a rooted tree. For any maximal clique C of G, let $G(C)$ be the subgraph induced by $V(C)$, which is the union of all maximal cliques in the subtree of T rooted at C.

The algorithm will calculate a minimum R-clique-transversal set of G by dynamic programming on T. For each maximal clique C and each subset S of C, let $CT_R(G(C), S)$ be a minimum sized R-clique-transversal set of $G(C)$ that contains S. In the following lemma, for any two sets X and Y, MIN $\{X,Y\}$ is equal to X if $|X| \leqslant |Y|$ and Y otherwise.

Lemma 3. *Suppose C_i is a maximal clique of G, whose children in T are C_{i_1}, $C_{i_2}, \ldots, C_{i_s}$. For any subset S of C_i,*

$$CT_R(G(C_i), S) = \begin{cases} \mathrm{MIN}\{X, Y\} & \text{if } |S| \geqslant r_i, \\ Y & \text{otherwise,} \end{cases}$$

where

$$X = S \cup \bigcup_{j=1}^{s} CT_R(G(C_{i_j}), S \cap C_{i_j}),$$

$$Y = \mathrm{MIN}\{CT_R(G(C_i), S \cup \{x\}) : \ x \in C_i - S\}.$$

Proof. Let D^* be the set at the right hand side of the above formula. It is clear that if $|S| \geqslant r_i$, then $X = S \cup \bigcup_{j=1}^{s} CT_R(G(C_{i_j}), S \cap C_{i_j})$ is an R-clique-transversal set of $G(C_i)$ that contains S. For any $x \in C_i - S$, it is also the case that $CT_R(G(C_i), S \cup \{x\})$ is an R-clique-transversal set of $G(C_i)$ that contains S. So D^* is an R-clique-transversal set of $G(C_i)$ that contains S.

On the other hand, suppose D is a minimum R-clique-transversal set of $G(C_i)$ that contains S. If $D \cap C_i$ contains some vertex $x \in C_i - S$, then D is an R-clique-transversal set of $G(C_i)$ that contains $S \cup \{x\}$. In this case,

$$|D| \geqslant |CT_R(G(C_i), S \cup \{x\})| \geqslant |Y| \geqslant |D^*|.$$

If $D \cap C_i = S$, then $|S| \geqslant r_i$ and each $D \cap V(C_{i_j})$ is an R-clique-transversal set of $G(C_{i_j})$ that contains $S \cap C_{i_j}$ for $1 \leqslant j \leqslant s$. In this case,

$$\begin{aligned} |D| &= |S| + \sum_{j=1}^{s} |(D \cap V(C_{i_j})) - S| \\ &= |S| + \sum_{j=1}^{s}(|D \cap V(C_{i_j})| - |C_{i_j} \cap S|) \\ &\geqslant |S| + \sum_{j=1}^{s}(|CT_R(G(C_{i_j}), S \cap C_{i_j})| - |C_{i_j} \cap S|) \\ &= |S| + \sum_{j=1}^{s} |CT_R(G(C_{i_j}), S \cap C_{i_j}) - S| \\ &\geqslant |S \cup \bigcup_{j=1}^{s} CT_R(G(C_{i_j}), S \cap C_{i_j})| \\ &= |X| \geqslant |D^*|. \end{aligned}$$

Therefore, D^* is a minimum R-clique-transversal set of $G(C_i)$ that contains S. □

The above lemma allows us to find a minimum R-clique-transversal set of a k-tree G recursively from the leaves to the root of T. For the case where C_i is a leaf, $s = 0$. When the solutions for all children of a maximal clique C_i are found, we compute $CT_R(G(C_i), S)$ for all $S \subseteq C_i$ in nonincreasing order of $|S|$ so that in calculating $CT_R(G(C_i), S)$ we may assume that $CT_R(G(C_i), S')$ is known for all S' with $|S'| > |S|$. Note that the algorithm is exponential in k, but linear when k is bounded.

Theorem 4. *The generalized clique-transversal problem can be solved in linear time for k-trees with fixed k.*

Corollary 5. *The clique-transversal problem, the $C_{i,1}$ problem, and the maximum q-colorable subgraph problem can be solved in linear time for k-trees with fixed k.*

Note that precisely the same argument also solves the above problems in linear time on chordal graphs with bounded clique sizes.

5. NP-complete results

This section establishes NP-complete results on k-tree (with unbounded k) and undirected path graphs. Besides clique-transversal, we also deal with two related concepts: domination and neighborhood-covering.

The concept of domination provides a natural model for many location problems in operations research. In a graph $G = (V, E)$, a vertex u is said to *dominate* a vertex v if $u \in N[v]$. A *dominating set* of G is a subset D of V such that every vertex in V is dominated by some vertex in D. The *domination number* $\gamma(G)$ of G is the minimum size of a dominating set of G. A *k-independent set* is a vertex subset of V such that the distance between any two distinct vertices is greater than k. 1-independence is the normal independence. The *k-independence number* $\alpha_k(G)$ of G is the maximum size of a k-independent set of G. 2-independence is a natural dual of domination. We also have a weak duality inequality: for any graph G,

$$\alpha_2(G) \leqslant \gamma(G). \tag{3}$$

Domination has been studied extensively by many authors during the past two decades (see [25]).

The concept of neighborhood-covering was first introduced by Sampathkumar and Neeralagi [34] and then studied by [5,27,30]. This is a vertex-edge variation of the domination problem. A vertex is said to *dominate* an edge if it dominates both end vertices of the edge. A *neighborhood-covering set* is a subset D of V such that every edge or vertex of G is dominated by some vertex in D. (This definition is slightly different from that in [5,30] where isolated vertices are not required to be dominated by D.) The *neighborhood-covering number* $\rho_N(G)$ is the minimum size of a neighborhood-covering set of G. A *neighborhood-independent set* is a set of edges and isolated vertices, which are viewed as edges with two identical end vertices, such that no two distinct edges in the set are dominated by a same vertex in V. The *neighborhood-independence number* $\alpha_N(G)$ is the maximum size of a neighborhood-independent set of G. For any graph G,

$$\alpha_N(G) \leqslant \rho_N(G). \tag{4}$$

It is NP-complete to determine $\alpha_N(G)$ and $\rho_N(G)$ for split graphs G [5]. There are linear algorithms for finding $\alpha_N(G)$ and $\rho_N(G)$ of an interval graph G [30] and a strongly chordal graph G [5]. These problems have been generalized to k distance version in [27].

Besides the weak duality inequalities (2) – (4), these six parameters are related by the following inequalities: for any graph G,

$$\gamma(G) \leqslant \rho_N(G) \leqslant \tau_C(G) \quad \text{and} \quad \alpha_2(G) \leqslant \alpha_N(G) \leqslant \alpha_C(G). \tag{5}$$

The first inequality follows from the fact that a neighborhood-covering set is a dominating set. The second inequality follows from the fact that a clique-transversal set is a neighborhood-covering set, since each edge or vertex is contained in a maximal clique. The third inequality follows from the fact that replacing each nonisolated vertex of a 2-independent set by an edge incident to it results in a neighborhood-independent set. The last inequality follows from the fact that replacing each edge or vertex of a neighborhood-independent set by a maximal clique containing it yields a clique-independent set. As a by-product of the linear time algorithms [5] for determining $\rho_N(G)$, $\tau_C(G)$, $\alpha_N(G)$, and $\alpha_C(G)$ of a strongly chordal graph G, these four parameters are equal for a strongly chordal graph.

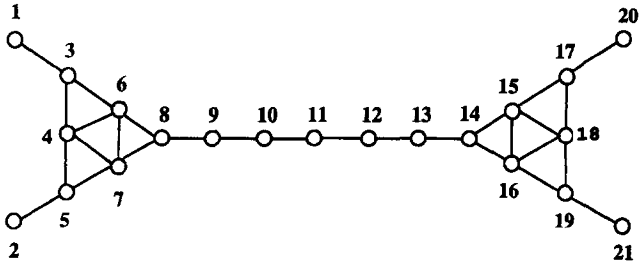

Fig. 1. A chordal graph G.

The inequalities in (5) can be strict even for chordal graphs. Fig. 1 shows an example of such a chordal graph:

$$\begin{aligned}
\gamma(G) &= 7, & D^* &= \{3, 5, 8, 11, 14, 17, 19\};\\
\rho_N(G) &= 8, & D^* &= \{3, 5, 8, 10, 12, 14, 17, 19\};\\
\tau_C(G) &= 9, & D^* &= \{3, 5, 7, 9, 11, 13, 16, 17, 19\};\\
\alpha_2(G) &= 7, & S^* &= \{1, 2, 8, 11, 14, 20, 21\};\\
\alpha_N(G) &= 8, & S^* &= \{\{1,3\}, \{2,5\}, \{6,7\}, \{9,10\}, \{12,13\},\\
&&&\quad \{15,16\}, \{17,20\}, \{19,21\}\};\\
\alpha_C(G) &= 9, & S^* &= \{\{1,3\}, \{2,5\}, \{4,6,7\}, \{8,9\}, \{10,11\},\\
&&&\quad \{13,14\}, \{15,16,18\}, \{17,20\}, \{19,21\}\}.
\end{aligned}$$

However, for some subclasses of chordal graphs, the inequalities in (5) can be all equalities. This fact plays an important role in the reductions between NP-complete problems. We shall consider the following subclasses of chordal graphs: split graphs, k-trees with unbounded k, and undirected path graphs.

The following construction of G_1 is from [5] and [8] for proving NP-complete results on split graphs. It provides a standard model for our proofs of NP-complete results in k-tree (with unbounded k) and undirected path graphs.

For any hypergraph $H = (V,E)$, construct the graph $G_1 = (V_1,E_1)$ with $V_1 = V \cup E$ such that V is a clique in G_1, E is an independent set in G_1, and $v \in V$ is adjacent to $e \in E$ in G_1 if and only if $v \in e$.

The basic idea for the NP-complete results in [5] is that for any hypergraph H,

$$\tau(H) = \gamma(G_1) = \rho_N(G_1) = \tau_C(G_1),$$

$$m(H) = \alpha_2(G_1) = \alpha_N(G_1) = \alpha_C(G_1);$$

and by [8],

$$\tau(H) \leqslant k \quad \text{if and only if} \quad C_{k+1,1}(\overline{G}_1) \leqslant |E| - k.$$

On the other hand, it is NP-complete to determine $\tau(H)$ of a 2-uniform hypergraph H, which is known as the "vertex cover" or the "hitting set" problem (see [15]). It is also NP-complete to determine $m(H)$ of a 3-partite 3-uniform hypergraph H, which is called the "three-dimensional matching" problem (see [15]). Hence we have the following results.

Theorem 6 ([5]). *It is* NP-*complete to determine* $\gamma(G)$, $\rho_N(G)$, *and* $\tau_C(G)$ (*resp.* $\alpha_2(G)$, $\alpha_N(G)$, *and* $\alpha_C(G)$) *of a split graph G with only degree*-2 (*resp. degree*-3) *vertices in the independent set.*

Theorem 7 ([8]). *The $C_{i,1}$ (and hence the maximum q-colorable subgraph) problem is* NP-*complete on split graphs.*

A similar idea was used in [9] to prove the NP-completeness of the domination problem on k-trees with unbounded k. Extending the argument, we have Theorem 8.

Theorem 8. *It is* NP-*complete to determine* $\gamma(G)$, $\rho_N(G)$, $\tau_C(G)$, $\alpha_2(G)$, $\alpha_N(G)$, *and* $\alpha_C(G)$ *of a k-tree G with unbounded k.*

Proof. For a hypergraph $H=(V,E)$, we construct an n-tree $G_2=(V_2,E_2)$ as follows. For each edge e of H, order the vertices of H into $v_{e,1}$, $v_{e,2},\ldots,v_{e,n}$ such that $e=\{v_{e,n+1-|e|}, v_{e,n-|e|},\ldots,v_{e,n}\}$. Now define the graph $G_2=(V_2,E_2)$ as

$$\begin{aligned} V_2 &= V\cup\{e^{(i)}: e\in E \text{ and } 1\leqslant i\leqslant n+1-|e|\},\\ E_2 &= \{uv : u\neq v \text{ in } V\}\\ &\quad\cup\{e^{(i)}e^{(j)}: e\in E \text{ and } 1\leqslant j<i\leqslant n+1-|e|\}\\ &\quad\cup\{e^{(i)}v_{e,j}: e\in E, 1\leqslant i\leqslant n+1-|e|, \text{ and } i\leqslant j\leqslant n\}. \end{aligned}$$

It is clear that G_2 is an n-tree. Fig. 2 shows an example of G_2, where H is a 2-uniform hypergraph of four vertices. Note that $e^{(n+1-|e|)}$ in G_2 plays the same role as e in G_1. G_2 is in fact G_1 with more $e^{(i)}$ added to make it an n-tree. The theorem follows from the following claim.

Claim. *For any hypergraph H, $\tau(H)=\gamma(G_2)=\rho_N(G_2)=\tau_C(G_2)$ and $m(H)=\alpha_2(G_2)=\alpha_N(G_2)=\alpha_C(G_2)$.*

Proof. Suppose D is a minimum dominating set of G_2. For each $v\in V$, $e\in E$ with $v\in e$, $1\leqslant i\leqslant n+1-|e|$, replace each $e^{(i)}\in D$ by $v_{e,n}$ since $v_{e,n}$ dominates the vertices adjacent to $e^{(i)}$, thus we may assume that $D\subseteq V$. For any $e\in E$, since D is a dominating set of G_2, $e^{(n+1-|e|)}$ is adjacent to some $v_{e,j}\in D$ with $n+1-|e|\leqslant j\leqslant n$; i.e., e meets D at $v_{e,j}$ in H. Then D is a transversal set of H. Thus $\tau(H)\leqslant\gamma(G_2)$. On the other hand,

$$C(e,i)=\{e^{(j)}:1\leqslant j\leqslant i\}\cup\{v_{e,j}: i\leqslant j\leqslant n\},$$

$e\in E$ and $1\leqslant i\leqslant n+1-|e|$, are precisely all maximal cliques of G_2. Suppose D is a minimum transversal set of H. For any maximal $C(e,i)$ of G_2, there exists $v_{e,j}\in e\cap D$ for some $n+1-|e|\leqslant j\leqslant n$. Since $i\leqslant n+1-|e|$, $i\leqslant j\leqslant n$ and then $v_{e,j}\in C(e,i)$. Thus D is a clique-transversal set of G_2. This gives $\tau_C(G_2)\leqslant\tau(H)$. These two inequalities and (5) together imply the first part of the claim.

For any matching M of H, $\{e^{(n+1-|e|)}: e\in M\}$ is clearly a 2-independent set of G_2 with the same size as M. So $m(H)\leqslant\alpha_2(G_2)$. On the other hand, suppose $\mathscr{P}$ is a

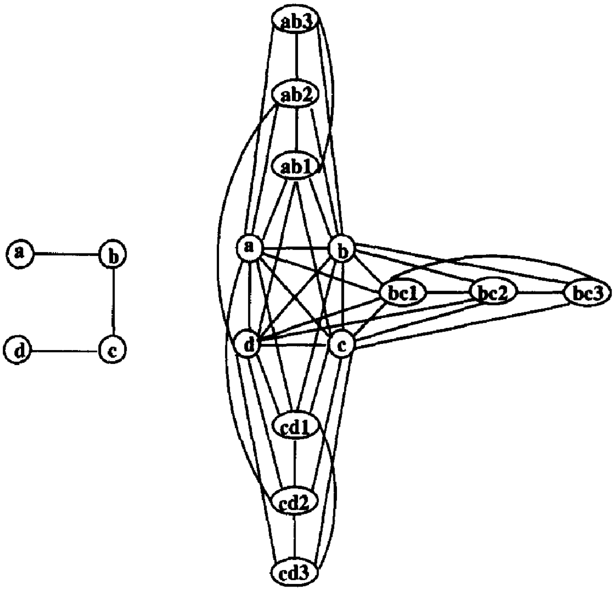

Fig. 2. A 2-uniform hypergraph H of four vertices and its G_2.

maximum clique-independent set of G_2. We may assume that $\mathscr{P}$ contains only maximal cliques of the type $C(e, n+1-|e|)$ with $e \in E$. $\{e \in E : C(e, n+1-|e|) \in \mathscr{P}\}$ is then a matching of H with the same size as $\mathscr{P}$. Hence, $\alpha_C(G_2) \leqslant m(H)$. These two inequalities together with (5) imply the second part of the claim. □

Note that all maximal cliques of G_2 are of size $n+1$. So the clique-transversal problem is the same as the $C_{n+1,1}$ problem in G_2.

Corollary 9. *The generalized clique-transversal problem, the $C_{i,1}$ problem with nonfixed i, and the maximum q-colorable subgraph problem with nonfixed q are* NP-*complete on k-trees with unbounded k.*

[4] proved that the domination problem is NP-complete on undirected path graphs by reducing the three-dimensional matching (3DM) problem to it. Extending the argument, we have the following theorem.

Theorem 10. *The clique-transversal problem and the neighborhood covering problem are* NP-*complete on undirected path graphs.*

Proof. Consider an instance of the 3DM problem, in which there are three disjoint sets W, X, and Y, each of cardinality q, and a subset M of $W \times X \times Y$ having cardinality p, say,

$$M = \{m_i = (w_r, x_s, y_t) : w_r \in W, x_s \in X, y_t \in Y, 1 \leqslant i \leqslant p\}.$$

The problem is to find a subset M' of M having cardinality exactly q such that each $w_r \in W$, $x_s \in X$, and $y_t \in Y$ occurs precisely once in a triple of M'.

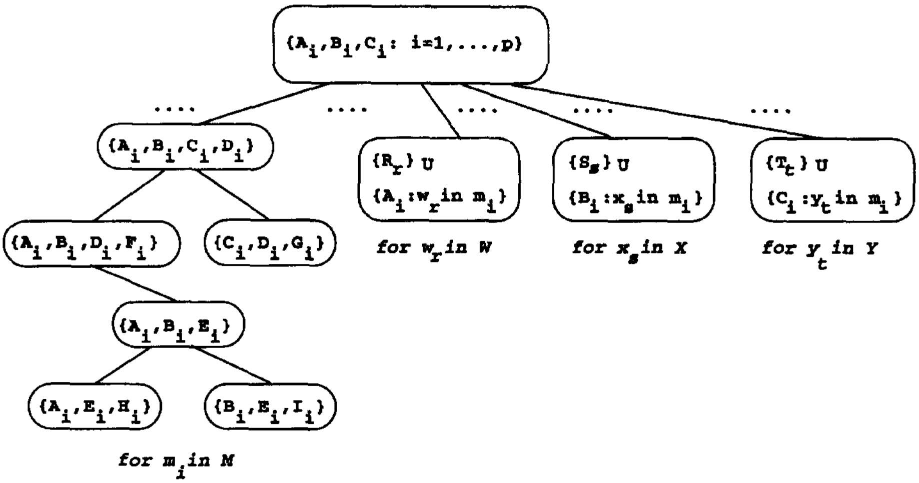

Fig. 3. A clique tree of G_3.

Consider the 3-partite 3-uniform hypergraph $H = (W \cup X \cup Y, M)$. Note that $m(H) \leqslant q$. The 3DM problem is equivalent to asking if $m(H) = q$.

Given an instance of the 3DM problem, or, equivalently, the corresponding hypergraph H, we construct a clique tree having $6p + 3q + 1$ cliques from which we obtain an undirected path graph G_3. The maximal cliques of the tree are explained below.

For each triple $m_i \in M$ there are six cliques whose vertices depend only upon the triple itself and not upon the elements within the triple: $\{A_i, B_i, C_i, D_i\}$, $\{A_i, B_i, D_i, F_i\}$, $\{C_i, D_i, G_i\}$, $\{A_i, B_i, E_i\}$, $\{A_i, E_i, H_i\}$, $\{B_i, E_i, I_i\}$ for $1 \leqslant i \leqslant p$. These six cliques form the subtree corresponding to m_i, which is illustrated in Fig. 3. Next, there is a clique for each element of W, X, and Y that depends upon the triples of M to which each respective element belongs:

$$\{R_r\} \cup \{A_i : w_r \in m_i\} \quad \text{all } w_r \in W,$$
$$\{S_s\} \cup \{B_i : x_s \in m_i\} \quad \text{all } x_s \in X,$$
$$\{T_t\} \cup \{C_i : y_t \in m_i\} \quad \text{all } y_t \in Y.$$

Finally, there is one large clique, the root of the clique tree, which contains $\{A_i, B_i, C_i : 1 \leqslant i \leqslant p\}$. The arrangement of these cliques is shown in Fig. 3. Note that G_3 has $9p + 3q$ vertices: $A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, I_i$ $(1 \leqslant i \leqslant p)$, R_r $(1 \leqslant r \leqslant q)$, S_s $(1 \leqslant s \leqslant q)$, T_t $(1 \leqslant t \leqslant q)$. The undirected path corresponding to a vertex v of G_3 consists of those cliques containing v in the clique tree. The theorem follows from the following claim.

Claim. *For the* 3*-partite* 3*-uniform hypergraph H corresponding to the* 3*DM problem, $m(H) = q$ if and only if $\gamma(G_3) = \rho_N(G_3) = \tau_C(G_3) = 2p + q$.*

Proof. Suppose D is a minimum dominating set of G_3. Observe that for any i, the only way to dominate the subtree corresponding to m_i with two vertices is to choose

D_i and E_i, and that any larger dominating set might just as well consist of A_i, B_i, and C_i, since none of the other possible vertices dominate any vertex outside of the subtree. Consequently, D consists of A_i, B_i, and C_i for t m_i's, and D_i and E_i for $p-t$ other m_i's, and at least $\max\{3(q-t),0\}$ R_r, S_s, T_t. Then

$$|D| \geqslant 3t + 2(p-t) + \max\{3(q-t),0\} \geqslant 2p + q + \max\{2(q-t), t-q\} \geqslant 2p + q.$$

This together with (5) gives $2p+q \leqslant \gamma(G_3) \leqslant \rho_N(G_3) \leqslant \tau_C(G_3)$.

For the case of $\gamma(G_3) = \rho_N(G_3) = \tau_C(G_3) = 2p+q$, the above minimum dominating set D has $t = q$ and consists of precisely A_i, B_i, and C_i for q m_i's, and D_i and E_i for $p-q$ other m_i's. Since all R_r, S_s, T_t are dominated by D, the q triples m_i for which A_i, B_i, and C_i are in D form a matching of size q. So $m(H) = q$.

Conversely, suppose H has a maximum matching M' of size q. Let

$$D = \{A_i, B_i, C_i : m_i \in M'\} \cup \{D_i, E_i : m_i \in M - M'\}.$$

Then $|D| = 3q + 2(p-q) = 2p + q$. It is straightforward to check that D is a clique-transversal set of G_3. So $\tau_C(G_3) \leqslant 2p+q$ and then $\gamma(G_3) = \rho_N(G_3) = \tau_C(G_3) = 2p+q$. □

In the above construction of the clique tree, if we add to each clique some vertices that appear only in it so that each clique is of size i, then we have the following result.

Theorem 11. *The $C_{i,1}$ problem (and hence the maximum q-colorable subgraph problem) is* NP-*complete on undirected path graphs if i (q) is part of the input.*

Acknowledgements

The authors are grateful to the three anonymous referees for many constructive suggestions.

References

[1] T. Andreae, M. Schughart and Z. Tuza, Clique-transversal sets of line graphs and complements of line graphs, Discrete Math. 88 (1991) 11–20.

[2] C. Berge, Graphs and Hypergraphs (North-Holland, Amsterdam, 1973).

[3] A.A. Bertossi, Dominating sets for split and bipartite graphs, Inform. Process. Lett. 19 (1984) 37–40.

[4] K.S. Booth and J.H. Johnson, Dominating sets in chordal graphs, SIAM J. Comput. 11 (1982) 191–199.

[5] G.J. Chang, M. Farber and Z. Tuza, Algorithmic aspects of neighborhood numbers, SIAM J. Disc. Math. 6 (1993) 24–29.

[6] G.J. Chang and G.L. Nemhauser, The k-domination and k-stability problems on sun-free chordal graphs, SIAM J. Algebraic Discrete Methods 5 (1984) 332–345.

[7] M. Conforti, D.G. Corneil and A.R. Mahjoub, k,i-Covers I: complexity and polytopes, Discrete Math. 58 (1986) 121–142.

[8] D.G. Corneil and J. Fonlupt, The complexity of generalized clique covering, Discrete Appl. Math. 22 (1988/89) 109–118.

[9] D.G. Corneil and J.M. Keil, A dynamic programming approach to the dominating set problem on k-trees, SIAM J. Algebraic Discrete Methods 8 (1987) 535–543.

[10] P. Erdős, T. Gallai and Z. Tuza, Covering the cliques of a graph with vertices, Discrete Math., to appear.

[11] M. Farber, Characterizations of strongly chordal graphs, Discrete Math. 43 (1983) 173–189.

[12] S. Földes and P.L. Hammer, Split graphs, in: F. Hoffman et al., eds., Proceedings 8th Southeastern Conference on Combinatorics, Graph Theory and Computing, Louisiana State University, Baton Rouge, Louisiana (1977) 311–315.

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

[14] D.R. Fulkerson and O.A. Gross, Incidence matrices and interval graphs, Pacific J. Math. 15 (1965) 835–855.

[15] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness (Freeman, San Francisco, CA, 1979).

[16] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, SIAM J. Comput. 1 (1972) 180–187.

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

[18] F. Gavril, A recognition algorithm for the intersection graphs of directed paths in directed trees, Discrete Math. 13 (1975) 237–249.

[19] F. Gavril, A recognition algorithm for the intersection graphs of paths in trees, Discrete Math. 23 (1978) 221–227.

[20] F. Gavril, Algorithms for maximum k-colorings and k-coverings of transitive graphs, Networks 17 (1987) 465–470.

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

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

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

[24] A. Hajnal and J. Surányi, Über die Auflosung von Graphen in vollständige Teilgraphen, Ann. Univ. Sci. Budapest Eötvös Sect. Math. 1 (1958) 113–121.

[25] S.T. Hedetniemi and R.C. Laskar, Bibliography on domination in graphs and some definitions of domination parameters, Discrete Math. 86 (1990) 257–277.

[26] A.J. Hoffman, A.W.J. Kolen and M. Sakarovitch, Totally balanced and greedy matrices, SIAM J. Algebraic Discrete Methods 6 (1985) 721–730.

[27] S.H. Hwang and G.J. Chang, k-Neighborhood covering and independence problems, DIMACS Report 93-03, DIMACS Center, Rutgers University (1993).

[28] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, eds., Complexity of Computer Computation (Plenum, New York, 1972) 85–103.

[29] C.G. Lekkerkerker and J.C. Boland, Representation of a finite graph by a set of intervals on the real line, Fund. Math. 51 (1962) 45–64.

[30] J. Lehel and Z. Tuza, Neighborhood perfect graphs, Discrete Math. 61 (1986) 93–101.

[31] M.V. Marathe, R. Ravi and C. Pandu Rangan, Generalized vertex covering in interval graphs, Discrete Appl. Math. 39 (1992) 87–93.

[32] R. Paige and R.E. Tarjan, Three partition refinement algorithms, SIAM J. Comput. 16 (1987) 973–989.

[33] D.J. Rose, Triangulated graphs and the elimination process, J. Math. Anal. Appl. 32 (1970) 597–609.

[34] E. Sampathkumar and P.S. Neeralagi, The neighborhood number of a graph, Indian J. Pure Appl. Math. 16 (1985) 126–132.

[35] J.P. Spinrad, Doubly lexical ordering of dense 0-1 matrices, preprint.

[36] Z. Tuza, Covering all cliques of a graph, Discrete Math. 86 (1990) 117–126.

[37] M. Yannakakis, Edge-deletion problems, SIAM J. Comput. 10 (1981) 297–309.

[38] M. Yannakakis and F. Gavril, The maximum k-colorable subgraph problem for chordal graphs, Inform. Process. Lett. 24 (1987) 133–137.