International Scholarly Research Network
ISRN Discrete Mathematics
Volume 2011, Article ID 540834, 20 pages
doi:10.5402/2011/540834

Research Article

Weighted Maximum-Clique Transversal Sets of Graphs

Chuan-Min Lee

Department of Computer and Communication Engineering, Ming Chuan University, 5 De Ming Road, Guishan District, Taoyuan County 333, Taiwan

Correspondence should be addressed to Chuan-Min Lee, joneslee@mail.mcu.edu.tw

Received 16 November 2011; Accepted 26 December 2011

Academic Editors: E. Cheng, A. Pêcher, and E. Tomita

A *maximum-clique transversal set* of a graph G is a subset of vertices intersecting all maximum cliques of G. The *maximum-clique transversal set problem* is to find a maximum-clique transversal set of G of minimum cardinality. Motivated by the placement of transmitters for cellular telephones, Chang, Kloks, and Lee introduced the concept of maximum-clique transversal sets on graphs in 2001. In this paper, we study the *weighted version* of the maximum-clique transversal set problem for split graphs, balanced graphs, strongly chordal graph, Helly circular-arc graphs, comparability graphs, distance-hereditary graphs, and graphs of bounded treewidth.

1. Introduction

All graphs considered in this paper are undirected, finite, and simple. Let $G = (V, E)$ be a graph with vertex set V and edge set E. Unless stated otherwise, it is understood that $|V| = n$ and $|E| = m$. For a graph G, we also use $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. We use $G[W]$ to denote a subgraph of G *induced* by a subset W of V, that is, $G[W]$ is the subgraph with vertex set W in which two vertices are adjacent whenever they are adjacent in G. For any vertex $v \in V$, the *neighborhood* of v in G is $N_G(v) = \{u \in V \mid (u, v) \in E\}$ and the *closed neighborhood* of v in G is $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of a vertex v in G, denoted by $\deg_G(v)$, is the number of edges incident with v. If $\deg_G(v) = 0$, then v is an *isolated vertex* of G. A *clique* is a subset of pairwise adjacent vertices of V. A *maximal clique* is a clique that is not a proper subset of any other clique. We use $C(G)$ to denote the collection of all maximal cliques of G. A clique is *maximum* if there is no clique of G of larger cardinality. The *clique* *number* of G, denoted by $\omega(G)$, is the cardinality of a maximum clique of G. We use $Q(G)$ to denote the collection of all maximum cliques of G.

A *maximum-clique transversal set* of a graph $G = (V, E)$ is a subset of V intersecting all maximum cliques of G. The *maximum-clique transversal number* of G, denoted by $\tau_M(G)$, is the minimum cardinality of a maximum-clique transversal set of G. The *maximum-clique transversal set problem* is to find a maximum-clique transversal set of G of minimum cardinality. A *maximum-clique independent set* of G is a collection of pairwise disjoint maximum cliques of G. The *maximum-clique independence number*, denoted by $\alpha_M(G)$, is the maximum cardinality of a maximum-clique independent set of G. The *maximum-clique independent set problem* is to find a maximum-clique independent set of G of maximum cardinality.

Maximum-clique transversal sets were introduced by Chang et al. in 2001 [1]. One of the main objectives for their research on maximum-clique transversal sets is the placement of transmitter towers for cellular telephones. Chang et al. stated a cellular telephone tower placement problem as the maximum-clique transversal set problem. They considered the problem and presented fixed parameter and approximation results for planar graphs. They also investigated the problem for some other graph classes such as k-trees, strongly chordal graphs, graphs with few P_4s, comparability graphs, and distance-hereditary graphs. Recently, Lee [2] introduced some variations of the maximum-clique transversal set problem and presented complexity results for them on some well-known classes of graphs.

Maximum-clique transversal and maximum-clique independent sets are closely related to *clique transversal* and *clique independent* sets on graphs. A clique transversal set of a graph $G = (V, E)$ is a subset of V intersecting all *maximal* cliques of G and a clique independent set of G is a collection of pairwise disjoint *maximal* cliques of G. The clique transversal number of G, denoted by $\tau_C(G)$, is the minimum cardinality of a clique transversal set of G. The clique independence number of G, denoted by $\alpha_C(G)$, is the maximum cardinality of a clique independent set of G. The clique transversal (resp., independent) set problem is to find a clique transversal (resp., independent) set of G of minimum (resp., maximum) cardinality. The clique transversal and clique independent set problems have been widely studied in [1, 3–20].

In this paper, we study the *weighted version* of the maximum-clique transversal set problem. Let $w : V \to \mathbb{N}$ be a function assigning to each vertex v of $G = (V, E)$ a weight $w(v)$ such that all arithmetic operations on vertex weights can be performed in time $O(1)$. We call w a vertex-weight function and call $G = (V, E, w)$ a *weighted graph*. We let $w(S) = \sum_{u \in S} w(u)$ for any subset S of V and let $w(S)$ be the *weight* of S. The *weighted maximum-clique transversal set problem* is to find a maximum-clique transversal set S of a weighted graph $G = (V, E, w)$ such that $w(S)$ is minimized.

We present polynomial-time algorithms (most of them with linear running time) for the weighted maximum-clique transversal set problem on split graphs, balanced graphs, strongly chordal graphs, Helly circular-arc graphs, comparability graphs, distance-hereditary graphs, and graphs of bounded treewidth.

2. Split Graphs

In this section, we consider the weighted maximum-clique transversal set problem on split graphs.

Definition 2.1. A *split graph* is a graph $G = (I \cup Q, E)$, where the vertices of G can be partitioned into an independent set I and a clique Q.

Throughout this section, we use $G = (V, E, w)$ to denote a split graph G with a vertex-weight function w. Without loss of generality, we may assume that G has no isolated vertices and that the vertices of G have been partitioned into an independent set I and a *maximum* clique Q. We give Algorithm 1 to solve the weighted maximum-clique transversal set problem for a split graph $G = (I \cup Q, E, w)$.

Theorem 2.2. *Algorithm 1 finds a maximum-clique transversal set D of a split graph $G = (I \cup Q, E, w)$ of minimum weight in $O(n+m)$ time.*

Proof. The theorem holds trivially if G has only one maximum clique. We may assume that G has more than one maximum clique. We show the correctness of Algorithm 1 as follows.

Initially, $S = I$ and $w_N(v) = w(N_G(v))$ for each vertex $v \in Q$. At each iteration of Steps (5)–(12), the algorithm removes from S an element s with $\deg_G(s) < \omega(G) - 1$, and $w_N(v)$ is decreased by the weight $w(s)$ if v is adjacent to s. At the end of the last iteration of Steps (5)–(12), the set S consists of all vertices s with $\deg_G(s) = \omega(G) - 1$, and $w_N(v) = w(N_G(v) \cap Q) + w(N_G(v) \cap S)$ for each vertex $v \in Q$.

For every maximum clique of G other than Q, it has exactly one vertex in I and $\omega(G)-1$ vertices in Q. For a vertex $x \in I$ with $\deg_G(x) < \omega(G) - 1$, $N_G[x]$ is not a maximum clique of G. Therefore, $Q(G) = \{Q\} \cup \bigcup_{s \in S}\{N_G[s]\}$.

Assume that D' is a maximum-clique transversal set of G of minimum weight. Let x_1 and x_2 be two vertices in Q. For every maximum clique Q' of G other than Q, it has exactly one vertex in S and $\omega(G) - 1$ vertices in Q. It is clear that Q' contains at least one vertex in $\{x_1, x_2\}$ and thus D' contains at most two vertices in Q.

Let v_1 be a vertex in Q such that $w(v_1) = \min\{w(v) \mid v \in Q\}$. Let v_2 be a vertex in $Q \setminus \{v_1\}$ such that $w(v_2) = \min\{w(v) \mid v \in Q \setminus \{v_1\}\}$ and let v_3 be a vertex in Q such that $w(S \cup Q) - w_N(v_3) = \min\{w(S \cup Q) - w_N(v) \mid v \in Q\}$. We consider the following two cases.

Case 1 ($|D' \cap Q| = 2$). It can be easily verified that the set $\{v_1, v_2\}$ is a maximum-clique transversal set of G of minimum weight, and thus $w(D') = w(v_1) + w(v_2)$.

Case 2 ($|D' \cap Q| = 1$). Let $D' \cap Q = \{v'\}$ and let $D^* = \{v'\} \cup (S \setminus N_G(v'))$. If $S \setminus N_G(v') = \emptyset$, then v' is adjacent to every vertex in S. We have $w(D') = w(D^*) = w(v')$. Suppose that $S \setminus N_G(v') \neq \emptyset$. For any vertex $x \in S \setminus N_G(v')$, the maximum clique $N_G[x]$ does not contain the vertex v'. Since D' does not contain any vertex in $Q \setminus \{v'\}$, x must be included in D'. In other words, the set $S \setminus N_G(v')$ is a subset of D'. Then, the set $D^* = \{v'\} \cup (S \setminus N_G(v'))$ is a maximum-clique transversal set of G and $w(D') = w(D^*)$. Note that $S \cap Q = \emptyset$ and $w_N(v') = w(N_G(v') \cap Q) + w(N_G(v') \cap S)$. We have

$$\begin{aligned} w(S \cup Q) - w_N(v') &= w(S) + w(Q) - (w(N_G(v') \cap Q) + w(N_G(v') \cap S)) \\ &= (w(S) - w(N_G(v') \cap S)) + (w(Q) - w(N_G(v') \cap Q)) \\ &= w(S \setminus N_G(v')) + w(v') = w(D^*). \end{aligned} \tag{2.1}$$

Since $w(S \cup Q) - w_N(v_3) = \min\{w(S \cup Q) - w_N(v) \mid v \in Q\}$, the set $D = \{v_3\} \cup (S \setminus N_G(v_3))$ is a maximum-clique transversal set of G of minimum weight. Following the discussion above, the algorithm is correct.

Clearly, Step (1) and Steps (13)–(21) of Algorithm 1 can be done in $O(n)$ time. Steps (2)–(4) and Steps (5)–(12) can be done in $O(\sum_{v \in V(G)}(\deg_G(v) + 1)) = O(n+m)$ time. Hence, the running time of the algorithm is $O(n+m)$ time. □

```
Input: A split graph $G = (I \cup Q, E, w)$.
Output: A maximum-clique transversal set $D$ of $G$ of minimum weight.
$S = I$;
for each vertex $v \in Q$ do
    $w_N(v) = w(N_G(v))$;
end for
for each vertex $s \in I$ do
    if $\deg_G(s) < \omega(G) - 1$ then
        $S \leftarrow S \setminus \{s\}$;
        for each $v \in N_G(s)$ do
            $w_N(v) = w_N(v) - w(s)$;
        end for
    end if
end for
Find a vertex $v_1 \in Q$ such that $w(v_1) = \min\{w(v) \mid v \in Q\}$;
Find a vertex $v_2 \in Q \setminus \{v_1\}$ such that $w(v_2) = \min\{w(v) \mid v \in Q \setminus \{v_1\}\}$;
Find a vertex $v_3 \in Q$ such that $w(S \cup Q) - w_N(v_3) = \min\{w(S \cup Q) - w_N(v) \mid v \in Q\}$;
if $\min\{w(v_1) + w(v_2), w(S \cup Q) - w_N(v_3)\} = w(v_1) + w(v_2)$ then
    $D = \{v_1, v_2\}$;
else
    $D = \{v_3\} \cup (S \setminus N_G(v_3))$;
end if
Output the set $D$;
```

Algorithm 1: Finding a maximum-clique transversal set of a split graph of minimum weight.

3. Balanced Graphs

In this section, we consider the weighted maximum-clique transversal set problem on balanced graphs.

Let G be a graph. Suppose that $V(G) = \{v_1, v_2, \ldots, v_n\}$, $C(G) = \{C_1, C_2, \ldots, C_p\}$, and $Q(G) = \{Q_1, Q_2, \ldots, Q_\ell\}$. A *clique matrix* (resp., *maximum-clique matrix*) of G, is the (0,1)-matrix whose entry (i, j) is 1 if $v_j \in C_i$ (resp., $v_j \in Q_i$), and 0 otherwise. A (0,1)-matrix is *balanced* if it does not contain the vertex-edge incidence matrix of an odd cycle as a submatrix, or equivalently, if it does not contain a square submatrix of odd order with exactly two ones per row and column. A (0, 1)-matrix is *totally balanced* if it does not contain the vertex-edge incidence matrix of a cycle as a submatrix [21]. A graph G is *balanced* (resp., *totally balanced*) if a clique matrix of G is balanced (resp., totally balanced).

Balanced graphs have been considered in [2, 8, 12, 15, 22]. It can be easily verified that if a clique matrix of a graph G is balanced (resp., totally balanced), then all clique matrices of G are balanced (resp., totally balanced). Note that a maximum-clique matrix of a graph G is a submatrix of some clique matrix of G. By definition, a submatrix of a balanced matrix is also balanced. We have the following lemma.

Lemma 3.1. *If a graph G is balanced, then any maximum-clique matrix of G is balanced.*

Let 1 (resp., 0) be a vector with n 1's (resp., 0's) and let $x = (x_1, x_2, \ldots, x_n)$ be a column vector. Let $G = (V, E, w)$ be a weighted graph with $V = \{v_1, v_2, \ldots, v_n\}$. Suppose that M is a maximum-clique matrix of G. The weighted maximum-clique transversal set problem for G can be formulated as the following integer linear programming problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} w(v_i) \cdot x_i \\ \text{subject to} \quad & Mx \geq 1 \\ & x_i = 1 \text{ or } 0 \qquad \text{for } i = 1, 2, \ldots, \text{n}. \end{aligned} \tag{3.1}$$

Fulkerson et al. proved the following important property of balanced matrices.

Theorem 3.2 (Fulkerson et al. [22]). *If A is a balanced matrix, then the polyhedra $P_1(A) = \{x \mid Ax \geq 1, x \geq 0\}$ and $P_2(A) = \{x \mid Ax \leq 1, x \geq 0\}$ have only integer extreme points.*

Theorem 3.3. *For any weighted balanced graph G, the weighted maximum-clique transversal set problem can be solved in polynomial time.*

Proof. Balanced graphs are a subclass of *hereditary clique-Helly graphs* [8]. Prisner [23] showed that no connected hereditary clique-Helly graphs has more maximal cliques than edges. Then, a connected balanced graph has $O(m)$ maximal cliques. Since all maximal cliques of a hereditary clique-Helly graph can be enumerated in polynomial time by the algorithms in [24], all the maximum cliques can be extracted in polynomial time. Therefore, a maximum-clique matrix M of G can be computed in polynomial time.

Note that if the extreme points of the polyhedra defined by the linear relaxation of an integer linear programming problem are integers, then the optimal solution of the integer linear programming problem is equal to the optimal solution of its linear relaxation. It is wellknown that linear programming problems can be solved in polynomial time. Following Lemma 3.1 and Theorem 3.2, the weighted maximum-clique transversal set problem is polynomial-time solvable for balanced graphs. □

4. Strongly Chordal Graphs

In this section, we consider the weighted maximum-clique transversal set problem for strongly chordal graphs.

Let $G = (V, E)$ be a graph. A vertex v is *simplicial* if all vertices of $N_G[v]$ form a clique. The ordering $(v_1, v_2, \ldots, v_n)$ of the vertices of V is a *perfect elimination ordering* of G if for all $i \in \{1, \ldots, n\}$, v_i is a simplicial vertex of the subgraph G_i of G induced by $\{v_i, v_{i+1}, \ldots, v_n\}$. Let $N_i[v]$ denote the closed neighborhood of v in G_i. Rose [25] showed the characterization that a graph is *chordal* if and only if it has a perfect elimination ordering. A perfect elimination ordering is called a *strong elimination ordering* if the following condition is satisfied.

For $i \leq j \leq k$ if v_j and v_k belong to $N_i[v_i]$ in G_i, then $N_i[v_j] \subseteq N_i[v_k]$.

Farber [26] showed that a graph is *strongly chordal* if and only if it admits a strong elimination ordering. So far, the fastest algorithm to recognize a strongly chordal graph and give a strong elimination ordering takes $O(m \log n)$ [27] or $O(n^2)$ time [28].

Definition 4.1. Let $G = (V, E)$ be a graph and $Q(G) = \{Q_1, Q_2, \ldots, Q_\ell\}$. Let $X(G) = \bigcup_{i=1}^{\ell} Q_i$. The *$\mathcal{V}Q$-incidence graph* of G, denoted by $\mathscr{H}'(G)$, is defined as follows. The vertex set of $\mathscr{H}'(G)$ is $X(G) \cup S'(G)$ where $S'(G) = \{s'_1, s'_2, \ldots, s'_\ell\}$. In $\mathscr{H}'(G)$, (1) $S'(G)$ is an independent set, (2) two vertices of $X(G)$ are adjacent if they are adjacent in G, and (3) for $1 \le i \le \ell$, $s'_i \in S'(G)$ is adjacent to $v_j \in X(G)$ if $v_j \in Q_i$ in G.

Definition 4.2. A *dominating set* S of a graph G is a subset of $V(G)$ such that $|S \cap N_G[v]| \ge 1$ for every vertex $v \in V(G)$. The *domination number* of G, denoted by $\gamma(G)$, is the minimum cardinality of a dominating set of G. The *weighted dominating set problem* is to find a dominating set S of a weighted graph $G = (V, E, w)$ such that $w(S)$ is minimized.

Lemma 4.3. *Let $G = (V, E, w)$ be a weighted graph. Let $w_{\mathscr{H}'}$ be a vertex-weight function of $\mathscr{H}'(G)$ defined by $w_{\mathscr{H}'}(v) = \infty$ if $v \in S'(G)$ and $w_{\mathscr{H}'}(v) = w(v)$ if $v \in X(G)$. A maximum-clique transversal set of G of minimum weight is equivalent to a dominating set of $\mathscr{H}'(G)$ of minimum weight.*

Proof. Let $\widetilde{\tau}_M$ be the minimum weight of a maximum-clique transversal set of G and let $\widetilde{\gamma}$ be the minimum weight of a dominating set of $\mathscr{H}'(G)$.

Suppose that $\widehat{D}$ is a maximum-clique transversal set of G of minimum weight. Then every maximum clique of G has at least one vertex in $\widehat{D}$. By the construction of $\mathscr{H}'(G)$, a vertex $s' \in S'(G)$ is adjacent to the vertices of a maximum clique of G. Therefore, $\widehat{D}$ is a dominating set of $\mathscr{H}'(G)$. We have $\widetilde{\gamma} \le \widetilde{\tau}_M$.

Conversely, we let D be a dominating set of $\mathscr{H}'(G)$ of minimum weight. Clearly, D does not contain any vertex in $S'(G)$. For any vertex $s \in S'(G)$, the set D has a vertex in $N_{\mathscr{H}'(G)}(s)$. By the construction of $\mathscr{H}'(G)$, $N_{\mathscr{H}'(G)}(s)$ is a maximum clique of G. Therefore, D is a maximum-clique transversal of G. We have $\widetilde{\tau}_M \le \widetilde{\gamma}$.

Following the discussion above, $\widetilde{\tau}_M = \widetilde{\gamma}$ and thus the lemma holds. □

Lemma 4.4. *Let G be a strongly chordal graph. Then $\mathscr{H}'(G)$ is a strongly chordal graph, and a strong elimination ordering of $\mathscr{H}'(G)$ can be obtained from a strong elimination ordering of G in $O(n + m)$ time.*

Proof. A strongly chordal graph is chordal. It has at most n maximal cliques, and all of its maximal cliques can be enumerated in $O(n + m)$ time [29]. Then, all maximum cliques $Q_1, Q_2, \dots, Q_\ell$ of a strongly chordal graph can be enumerated in $O(n + m)$ time.

Let $C(G) = \{C_1, C_2, \dots, C_p\}$ and $S(G) = \{s_1, s_2, \dots, s_p\}$. The *vertex-clique incidence graph* of G, denoted by $\mathscr{H}(G)$, is defined as follows. The vertex set of $\mathscr{H}(G)$ is $V(G) \cup S(G)$. In $\mathscr{H}(G)$, (1) $S(G)$ is an independent set, (2) two vertices of V are adjacent if they are adjacent in G, and (3) for $1 \le i \le p$, $s_i \in S(G)$ is adjacent to $v_j \in V(G)$ if $v_j \in C_i$ in G. Let S' be a maximum subset of $S(G)$ such that $N_{\mathscr{H}(G)}(s')$ is a maximum clique of G for each vertex $s' \in S'$. Note that $Q(G) \subseteq C(G)$ and $X(G) \subseteq V(G)$. Therefore, $\mathscr{H}'(G)$ is isomorphic to the subgraph of $\mathscr{H}(G)$ induced by $X(G) \cup S'$.

Guruswami and Rangan [17] showed that $\mathscr{H}(G)$ is a strongly chordal graph with $O(n + m)$ edges and $O(n)$ vertices, and that a strong elimination ordering of $\mathscr{H}(G)$ can be obtained from a given one for G in $O(n + m)$ time. Therefore, $\mathscr{H}'(G)$ is also a strongly chordal graph and can be constructed from $\mathscr{H}(G)$ in $O(n + m)$ time.

Suppose that $n_1 = |V(\mathscr{H}(G))|$ and $n_2 = |V(\mathscr{H}'(G))|$. Then, $n_2 \le n_1$. Let $(w_1, w_2, \dots, w_{n_1})$ be a strong elimination ordering of $\mathscr{H}(G)$. Let $(w_{x_1}, w_{x_2}, \dots, w_{x_{n_2}})$ be the ordering of the vertices in $\mathscr{H}'(G)$ obtained by removing all the vertices in $V(\mathscr{H}(G)) \setminus V(\mathscr{H}'(G))$ from the ordering $(w_1, w_2, \dots, w_{n_1})$. It can be easily verified that the ordering $(w_{x_1}, w_{x_2}, \dots, w_{x_{n_2}})$ is also a strong elimination ordering of $\mathscr{H}'(G)$. Following the discussion above, the lemma holds. □

Theorem 4.5. *The weighted maximum-clique transversal set problem can be solved in $O(n+m)$ time for a strongly chordal graph G if a strong elimination ordering is given.*

Proof. It follows from Lemmas 4.3 and 4.4 and the result that the weighted dominating set problem can be solved in $O(n+m)$ time for a strongly chordal graph if a strong elimination ordering is given [30]. □

5. Helly Circular-Arc Graphs

In the section, we consider the weighted maximum-clique transversal set problem for Helly circular-arc graphs.

Definition 5.1. Let $\mathcal{S}$ be a set. Let $\mathcal{P}$ be a set of some positive integers. A collection $\{S_i\}_{i\in\mathcal{P}}$ of subsets of $\mathcal{S}$ is said to satisfy the *Helly property* if $J \subseteq \mathcal{P}$ and $S_i \cap S_j \neq \emptyset$ for all $i, j \in J$ implies $\bigcap_{j\in J} S_j \neq \emptyset$.

Let $\mathcal{F}$ be a collection of nonempty sets. The *intersection graph* G of $\mathcal{F}$ is obtained by representing each set in $\mathcal{F}$ as a vertex and connecting two vertices with an edge if and only if their corresponding sets intersect. A *circular-arc model* $\mathcal{M}$ is a pair $(\mathcal{C}, \mathcal{A})$, where $\mathcal{C}$ is a circle and $\mathcal{A}$ is a collection of arcs of $\mathcal{C}$. If a graph G is the intersection graph of $\mathcal{A}$, then G is a *circular-arc graph*. If $\mathcal{A}$ satisfies the Helly property, then G is a *Helly circular-arc graph* and $\mathcal{M}$ is called a *Helly circular-arc model* of G. For an arc $A \in \mathcal{A}$, let $v(A)$ be the vertex of G corresponding to A. For $\mathcal{A}' \subseteq A$, let $V(\mathcal{A}') = \bigcup_{A\in\mathcal{A}'}\{v(A)\}$. Let p be a point of $\mathcal{C}$ and let $\mathcal{A}(p)$ be the collection of arcs that contain p. If $V(\mathcal{A}(p))$ is a maximal clique of G, then p is called a *clique point*. Suppose that p_1 and p_2 are distinct points of $\mathcal{C}$. If $\mathcal{A}(p_1) = \mathcal{A}(p_2)$, then p_1 and p_2 are *equivalent*. A *clique point representation* of $\mathcal{M}$ is a maximum set of *nonequivalent* clique points of $\mathcal{C}$.

Let $\mathcal{M} = (\mathcal{C}, \mathcal{A})$ be a circular-arc model of a Helly circular-arc graph G. Let X be a maximal clique of G and let $\mathcal{W} = \{c_1, c_2, \ldots, c_p\}$ be a clique point representation of $\mathcal{M}$. Due to Helly property, the arcs corresponding to vertices in X have a point on the circle $\mathcal{C}$ in common. It is clear that a point b of $\mathcal{C}$ is a clique point if and only if $V(\mathcal{A}(b))$ is a maximal clique of G. Then, we have $C(G) = \{V(\mathcal{A}(c_1)), V(\mathcal{A}(c_2)), \ldots, V(\mathcal{A}(c_p))\}$.

If s and t are points of $\mathcal{C}$, we use (s,t) to denote an arc of $\mathcal{A}$ starting at s and ending at t in clockwise direction. For each arc $A = (s,t) \in \mathcal{A}$, the points s, t are called the *extremes* of A. We also use $s(A)$ and $t(A)$ to denote the *starting point* and *ending point* of the arc A, respectively. Without loss of generality, we assume that (1) all arcs of $\mathcal{C}$ are open arcs, (2) no single arc entirely covers $\mathcal{C}$, and (3) no two extremes of distinct arcs of $\mathcal{A}$ coincide.

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ and $A_i = (s_i, t_i)$ for $1 \leq i \leq n$. An *intersection segment* (s_i, t_j) is the contiguous part of $\mathcal{C}$ formed by two consecutive extremes s_i and t_j, where s_i is the starting point of some arc $A_i \in \mathcal{A}$ and t_j is the ending point of an arc $A_j \in \mathcal{A}$ in clockwise direction. A point inside an intersection segment is called an *intersection point*. There are at most n intersection segments and every clique point is an intersection point [13].

Let H be a weighted Helly circular-arc graph with a Helly circular-arc model $\mathcal{M} = (\mathcal{C}, \mathcal{A})$. Lin et al. [31] proposed a linear-time algorithm that finds a clique point representation $\mathcal{W}$ of $\mathcal{M}$. Based upon their algorithm, we have the following theorem.

Theorem 5.2. *Let $\mathcal{M} = (\mathcal{C}, \mathcal{A})$ be a Helly circular-arc model of a Helly circular-arc graph G. Let $\mathcal{W} = \{c_1, c_2, \ldots, c_p\}$ be a clique point representation of $\mathcal{M}$. Then, the clique number $\omega(G)$ and $\mathcal{A}(c_1), \mathcal{A}(c_2), \ldots, \mathcal{A}(c_p)$ can be computed in linear time.*

Guruswami and Rangan [17] showed that a Helly circular-arc graph G has at most n maximal cliques. The following lemma can be easily verified according to the Helly property.

Lemma 5.3. *Suppose that G is a Helly circular-arc graph with a Helly circular-arc model $\mathcal{M} = (\mathcal{C}, \mathcal{A})$ and $\omega(G) > 1$. Let v be a vertex of G. Then, there are at most $deg_G(v)$ maximal cliques of G containing the vertex v.*

Lemma 5.4. *Suppose that $G = (V, E)$ is a Helly circular-arc graph with a Helly circular-arc model $(\mathcal{C}, \mathcal{A})$.*

(1) *The $\mathcal{VQ}$-incidence graph $\mathcal{H}'(G)$ has $O(n+m)$ edges and $O(n)$ vertices.*

(2) *The $\mathcal{VQ}$-incidence graph $\mathcal{H}'(G)$ is a Helly circular-arc graph and a Helly circular-arc model $(\mathcal{C}, \mathcal{A}')$ of $\mathcal{H}'(G)$ can be obtained from $(\mathcal{C}, \mathcal{A})$ in $O(n+m)$ time.*

Proof. (1) By Definition 4.1, $V(\mathcal{H}'(G)) = X(G) \cup S'(G)$. Since G is a Helly circular-arc graph, it has at most n maximal cliques [17]. Then $\mathcal{H}'(G)$ has $O(n)$ vertices. Without loss of generality, we assume that the clique number $\omega(G) > 1$. By Lemma 5.3 and the construction of $\mathcal{H}'(G)$, every vertex $v \in X(G)$ is adjacent to at most $\deg_G(v)$ vertices in $S'(G)$. Hence, the number of $E(\mathcal{H}'(G))$ is $O(m + \sum_{v \in X(G)}(\deg_G(v) + 1)) = O(n+m)$.

(2) Let $\mathcal{W} = \{c_1, c_2, \ldots, c_p\}$ be a clique point representation of $\mathcal{M}$. The clique point representation $\mathcal{Q}$ can be constructed in linear time [31]. Following Theorem 5.2, the clique number $\omega(G)$ and the arc sets $\mathcal{A}(c_1), \mathcal{A}(c_2), \ldots, \mathcal{A}(c_p)$ can be computed in linear time. Then $C(G) = \{V(\mathcal{A}(c_1)), V(\mathcal{A}(c_2)), \ldots, V(\mathcal{A}(c_p))\}$.

Let $P = \{p_1, p_2, \ldots, p_\ell\}$ be a maximum subset of $\mathcal{Q}$ such that $V(\mathcal{A}(p_i))$ is a maximum clique of G for $1 \le i \le \ell$. The set P and the arc sets $\mathcal{A}(p_1), \mathcal{A}(p_2), \ldots, \mathcal{A}(p_\ell)$ can be computed in linear time. Then $Q(G) = \{V(\mathcal{A}(p_1)), V(\mathcal{A}(p_2)) , \ldots, V(\mathcal{A}(p_\ell))\}$.

Let $\mathcal{A}_P = \bigcup_{i=1}^{\ell} \mathcal{A}(p_i)$ and let A_W be a set of ℓ arcs of $\mathcal{C}$ such that each arc of $\mathcal{A}_W$ contains exactly one clique point of P and contains no extremes of arcs of $\mathcal{A}_P$. It follows that $\mathcal{H}'(G)$ is a Helly circular-arc graph and $(\mathcal{C}, \mathcal{A}_P \cup \mathcal{A}_W)$ is a Helly circular-arc graph model for $\mathcal{H}'(G)$. Hence, $\mathcal{H}'(G)$ is a Helly circular-arc graph, and a Helly circular-arc model $(\mathcal{C}, \mathcal{A}')$ of $\mathcal{H}'(G)$ can be obtained from $(\mathcal{C}, \mathcal{A})$ in $O(n+m)$ time. □

Theorem 5.5. *The weighted maximum-clique transversal set problem can be solved in $O(n+m)$ time for a Helly circular-arc graph $G = (V, E, w)$ if a Helly circular-arc model $\mathcal{M} = (\mathcal{C}, \mathcal{A})$ is given.*

Proof. It follows from Lemmas 4.3 and 5.4, and the result that the weighted dominating set problem can be solved in $O(n+m)$ time for a circular-arc graph if a circular-arc model is given [32]. □

6. Comparability Graphs

A *directed graph* (or just *digraph*) $D = (V, A)$ consists of a nonempty finite set V of vertices and a finite set A of ordered pairs of distinct vertices called *arrows*. We call V the *vertex set* and A the *arrow set* of D. We also use $V(D)$ and $A(D)$ to denote the vertex set and arrow set of D, respectively. For an arrow $\langle u, v \rangle$, the first vertex u is its *tail* and the second vertex v is its *head*. We also say that the arrow $\langle u, v\rangle$ *leaves* u and *enters* v. For a vertex v of D, we use the following notations:

$$N_D^+(v) = \{w \in V \setminus \{v\} \mid \langle v, w\rangle \in A\}, \qquad N_D^-(v) = \{u \in V \setminus \{\mathrm{v}\} \mid \langle u, v\rangle \in A\}. \tag{6.1}$$

The sets $N_D^+(v), N_D^-(v)$, and $N_D(v) = N_D^+(v) \cup N_D^-(v)$ are called the *outneighborhood*, *inneighborhood*, and *neighborhood* of v, respectively. We call the vertices in $N_D^+(v), N_D^-(v)$, and $N_D(v)$ the *outneighbors*, *inneighbors*, and *neighbors* of v, respectively.

A *directed walk* in a digraph $D = (V, A)$ from vertex u to vertex v, or simply a *directed* $(u,\ v)$*-walk* is a sequence of vertices $(v_0, v_1, \ldots, v_n)$ such that $u = v_0, v = v_n$, and $\langle v_{i-1}, v_i\rangle$ is an arrow in D for $1 \leq i \leq n$, where n is called the *length* of this walk. A *directed path* is a directed walk in which no vertex is repeated. A directed (s, t)*-path* is directed path starting at s and ending at t. A *directed Hamiltonian path* is a directed path that visits each vertex of D exactly once. A *directed cycle* is a directed (v, v)-walk in which no vertex is repeated except v. Arrow set A is a *transitive relation* on V if for all $u, v, w \in V$, the following holds:

$$\text{If } \langle u, v\rangle \in A \text{ and } \langle v, w\rangle \in A, \text{ then } \langle u, w\rangle \in A. \tag{6.2}$$

Let $G = (V, E)$ be an undirected graph. Then the directed graph $D = (V, A)$ is an *orientation* of G if for all $(x, y) \in E$, either $\langle x, y\rangle \in A$ or $\langle y, x\rangle \in A$ and for all $\langle x, y\rangle \in A$, $(x, y) \in E$ holds. If A is a transitive relation on V, then D is a *transitive orientation* of G. If there are no directed cycles in D, then D is an *acyclic orientation* of G. Assume that $D' = (V', A')$ is a directed graph. An undirected graph $G = (V', E)$ is the *underlying* graph of D' if for all $\langle x, y\rangle \in A'$, $(x, y) \in E$ and for all $(x, y) \in E$, either $\langle x, y\rangle \in A'$ or $\langle y, x\rangle \in A'$.

An undirected graph G is a comparability graph if and only if it has a transitive orientation. Figure 1 shows a comparability graph G and its transitive orientation.

Given a comparability graph $G = (V, E)$, a transitive orientation of G can be found in linear time [33]. Chang et al. [1] solved the maximum-clique transversal set problem in $O(m\sqrt{n})$ time for comparability graphs. In this section, we show how to use a transitive orientation of a comparability graph $G = (V, E)$ to solve the weighted maximum-clique transversal set problem on G in $O(nm \log(n^2/m))$ time.

Definition 6.1. A *tournament* is an orientation of a complete graph.

Theorem 6.2 (Rédei [34]). *Every tournament contains a directed Hamiltonian path.*

Lemma 6.3. *There exists a one-to-one correspondence between the set of maximum cliques of a comparability graph G and the set of longest directed paths of a transitive orientation D of G.*

Proof. Let $G = (V, E)$ be a comparability graph and let $D = (V, A)$ be a transitive orientation of G. By the transitive relation on V, each directed path in D corresponds to a clique of G. Let $(v_1, v_2, \ldots, v_k)$ be a longest directed path of D. If $S = \{v_1, v_2, \ldots, v_k\}$ is not a maximum clique of G, then there exists a vertex $v \in V \setminus S$ such that $S \cup \{v\}$ is a clique of G. With the help of Theorem 6.2, we know that there is a directed path of length $k+1$. However, it contradicts that the length of a longest directed path is k. Hence, each longest directed path of D corresponds to a maximum clique of G.

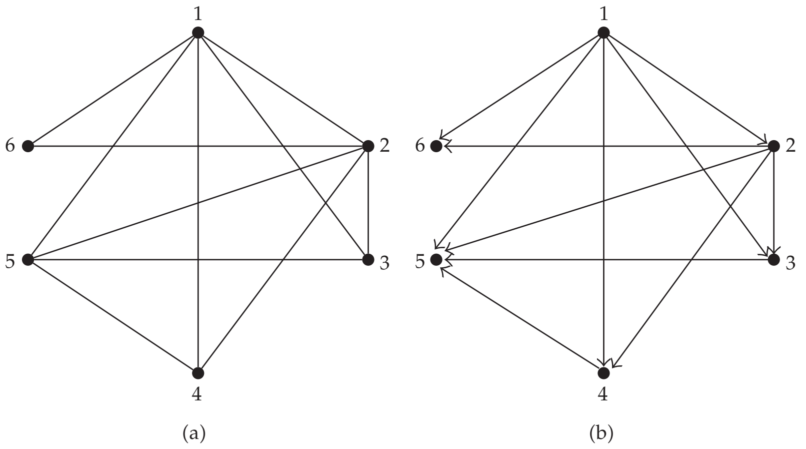


Figure 1: (a) A comparability graph G. (b) A transitive orientation of G.

Conversely, let $S = \{v_1, v_2, \ldots, v_k\}$ be a maximum clique of G. With the help of Theorem 6.2, we know that there exists a directed path P of length k. Assume that P is not a longest directed path in D. Then there exists a path of length greater than k. By the transitive relation on V, there is a clique whose number of vertices is larger than k. However, it contradicts that S is a maximum clique of G. Hence, each maximum clique of G corresponds to a longest directed path in D. Following the discussion above, the lemma holds. □

Suppose that $G = (V, E)$ is a comparability graph and $D = (V, A)$ is a transitive orientation of G. By the transitive relation on V, D has no directed cycle. It is known that an acyclic digraph H has a *topological sort* of $V(H)$, that is, a linear ordering of all vertices in $V(H)$ such that if H contains an arrow $\langle u, v\rangle \in A$, then u precedes v in the ordering [35]. Clearly, there exists at least one vertex u such that no vertex enters it. We call such vertices the *source vertices*. Similarly, there exists at least one vertex v such that no vertex leaves it. Such vertices are called the *sink vertices*. We add a new vertex s to D and add arrows from s to every source vertex in D. Correspondingly we add another new vertex t and arrows from every sink vertex in D to t. Let $D' = (V', A')$ be the resulting digraph. The digraph D' is called an (s, t)*-transitive orientation* of G.

Lemma 6.4. *Every longest directed path in a transitive orientation D of a comparability graph $G = (V, E)$ starts at a source vertex and ends at a sink vertex.*

Proof. It can be easily verified according to the transitive relation on V and the definition of a longest directed path. □

Lemma 6.5. *Let $G = (V, E)$ be a comparability graph and let D be a transitive orientation of G. Suppose that D' is the (s, t)-transitive orientation of G. Let $P = (s = v_0, v_1, v_2, \ldots, v_{i-1}, v_i = t)$ be an arbitrary directed (s, t)-path in D'. Directed path P has the longest length if and only if the path $(v_1, v_2, \ldots, v_{i-1})$ is a longest directed path in D.*

Proof. It can be easily verified by Lemma 6.4. □

Definition 6.6. Let $G = (V, E)$ be an undirected graph. A vertex set $S \subset V$ is a *vertex separator* of G if $G[V \setminus S]$ is disconnected. A set $S \subset V$ is an (a, b)*-vertex separator* of G if a and b are in distinct connected components of $G[V \setminus S]$.

In [1], Chang et al. remove some vertices (except for s and t) and arrows from an (s, t)-transitive orientation D' of a comparability graph G to obtain a new digraph D^* such that (1) every directed (s, t)-path in D^* is a longest directed path in D' and (2) every longest directed path in D' is a directed (s, t)-path in D^*. The construction of D^* can be done in linear time. We call D^* the (s, t)*-longest-path digraph* of G. By Lemmas 6.3 and 6.5, a subset S of $V(G)$ is a maximum clique if and only if s, t, and all vertices in S can form a directed (s, t)-path in D^*. Therefore, we have the following theorem.

Theorem 6.7. *Let G be a comparability graph with a vertex-weight function w and let D^* be an (s, t)-longest-path digraph of G. Suppose that G^* is the underlying graph of D^*. An (s, t)-vertex separator S of G^* of minimum weight $w(S)$ is equivalent to a maximum-clique transversal set of G of minimum weight.*

Definition 6.8. Let $D = (V, A)$ be a flow network with a capacity function c. Let s be the source vertex of the network, and let t be the sink vertex. Let S be a subset of vertices such that $s \in S$ and $t \in V \setminus S$. Let $\overline{S} = V \setminus S$. We use $(S; \overline{S})$ to denote the set of arrows which leave from a vertex of S and enter a vertex of $\overline{S}$. The $(S; \overline{S})$ is called a *cut* of D. Let $c(S) = \sum_{e \in (S;\overline{S})} c(e)$ be the *capacity of the cut* determined by S. A minimum cut S is a cut of D such that $c(S)$ is minimized.

Theorem 6.9 (Max-flow min-cut theorem [36, 37]). *In every network, the maximum total value of a flow equals the minimum capacity of a cut.*

Suppose that G is a comparability graph with a vertex-weight function w and D^* is an (s, t)-longest-path digraph of G. We construct a *flow network* $\widehat{D} = (\widehat{V}, \widehat{A})$ from D^* as follows:

(1) $\widehat{V} = \{s, t\} \cup \{v^1, v^2 \mid v \in V(D^*) \setminus \{s, t\}\}$;

(2) $\widehat{A} = A_1 \cup A_2 \cup A_3 \cup A_4$, where $A_1 = \{\langle s, v^1 \rangle \mid \langle s, v \rangle \in A(D^*)\}$, $A_2 = \{\langle v^1, v^2 \rangle \mid v \in V(D^*) \setminus \{s, t\}\}$, $A_3 = \{\langle u^2, v^1 \rangle \mid \langle u, v \rangle \in A(D^*)$ and $u, v \in V(D^*) \setminus \{s, t\}\}$, and $A_4 = \{\langle v^2, t \rangle \mid \langle v, t \rangle \in A(D^*)\}$;

(3) For each arrow $\langle v^1, v^2 \rangle \in A_2$, let the weight $w(v)$ be its capacity. For each edge in $A_1 \cup A_3 \cup A_4$, we assign the capacity ∞ to it.

Let $(S; \overline{S})$ be a minimum cut of $\widehat{D}$. By the *max-flow min-cut* theorem, $(S; \overline{S})$ does not contain any arrow in $A_1 \cup A_3 \cup A_4$. The set $(S; \overline{S})$ is a subset of A_2. Suppose that $(S; \overline{S}) = \{\langle x^1_{\ell_1}, x^2_{\ell_1} \rangle, \langle x^1_{\ell_2}, x^2_{\ell_2} \rangle, \ldots, \langle x^1_{\ell_i}, x^2_{\ell_i} \rangle\}$. Let $S' = \{x_{\ell_1}, x_{\ell_2}, \ldots, x_{\ell_i}\}$. It can be easily verified that S' is an (s, t)-vertex separator of G^* of minimum weight. Following Theorem 6.7, we know that the set S' is a maximal-clique transversal set of G of minimum weight. Note that a minimum cut of a flow network can be computed in $O(nm \log(n^2/m))$ time [38]. Therefore, we have the following result.

Theorem 6.10. *Given a comparability graph G with a vertex-weight function w, the weighted maximum-clique transversal set problem can be solved in $O(nm \log(n^2/m))$ time.*

7. Graphs of Bounded Treewidth k

In this section we show that the weighted maximum-clique transversal set problem can be solved in linear time for graphs of bounded treewidth.

A clique with k vertices is called a *k-clique*. A $(k+1)$-clique is a k-tree. A k-tree with $n+1$ vertices can be obtained from a k-tree with n vertices by making a new vertex adjacent to exactly all vertices of a k-clique. For a k-tree G, $\omega(G) = k$ if G is a k-clique and $\omega(G) = k+1$ otherwise. For convenience, we define k-trees as having at least $k+1$ vertices. With this definition, the treewidth of a k-tree is k and the clique number of a k-tree is $k+1$. Then, a k-clique is a $(k-1)$-tree and the treewidth of a k-clique is $k-1$.

Subgraphs of k-trees are called *partial k-trees*. If a partial k-tree G is a subgraph of a k-tree H, then we call H a *k-tree embedding* for G. The smallest k such that a graph G is a partial k-tree is called the treewidth of G. It is clear that a graph of treewidth k is also a partial ℓ-tree for every $\ell \geq k$. The class of partial k-trees is exactly the class of graphs of treewidth at most k.

The treewidth of a graph can be defined by the concept of *tree decompositions* of a graph (see, e.g., [39]).

Definition 7.1. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(T, \mathcal{S})$, where T is a tree with ℓ nodes and $\mathcal{S}$ is a collection of subsets $S_1, S_2, \ldots, S_\ell$ of V such that a node i in T corresponds to the subset $S_i \in \mathcal{S}$ for $1 \leq i \leq \ell$ and the following three conditions are satisfied.

(1) Every vertex $x \in V$ appears in at least one subset $S_i \in \mathcal{S}$.

(2) For every edge $e \in E$, there is at least one subset $S_i \in \mathcal{S}$ containing both endpoints of e.

(3) If a vertex x appears in two subsets $S_i, S_j \in \mathcal{S}$, then it appears in every subset S_k for k on the (unique) path from node i to node j in T.

Definition 7.2. The *width* of a tree decomposition $(T, \mathcal{S})$ of a graph G is the maximum cardinality *minus one* over all subsets of $\mathcal{S}$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G.

Lemma 7.3 (Bodlaender [40]). *If the treewidth of a graph $G = (V, E)$ is at most k, then $|E| \leq k|V| - (1/2)k(k+1)$.*

By Lemma 7.3, $O(|V| + |E|) = O(kn)$ for a partial k-tree $G = (V, E)$ with bounded k. It was shown in [40] that for *each* constant k it can be determined in linear time whether a graph G has treewidth at most k.

A tree decomposition $(T, \mathcal{S})$ is *rooted* if the tree T is equipped with some root node. A rooted tree decomposition is called *nice* if the following conditions are satisfied.

(1) Every node of T has at most two children.

(2) If a node i has two children j and k then $S_i = S_j = S_k$.

(3) If a node i has only one child j then either $|S_i| = |S_j| + 1$ and $S_j \subset S_i$ or $|S_i| = |S_j| - 1$ and $S_i \subset S_j$.

By [39], it is fairly easy to see that every graph with treewidth k has a nice tree decomposition of width k and that it can be obtained in linear time from an ordinary tree decomposition with the same width. Furthermore any graph on n vertices has a nice tree decomposition with at most $4n$ nodes. In the following, we assume that a nice tree decomposition $(T, \mathcal{S})$ of G of width k is part of the input.

Lemma 7.4 (Bodlaender and Möhring [41]). *Let $(T, \mathcal{S})$ be a tree decomposition for a graph G. For every clique C of G, there exists a subset $X \in \mathcal{S}$ such that C is a subset of X.*

Lemma 7.5. *Given a nice tree decomposition $(T, \mathcal{S})$ of a graph G of bounded treewidth k, the maximum cliques of $G = (V, E)$ can be enumerated in $O(3^{k/3}n)$ time.*

Proof. Note that a nice tree decomposition of a graph of bounded treewidth k has at most $4n$ nodes. In other words, $\mathcal{S}$ contains at most $4n$ subsets of V. Let $\mathcal{S} = \{S_1, S_2, \dots, S_\ell\}$. For $1 \le i \le \ell$, $|S_i| \le k + 1$. By Lemma 7.4, a maximum clique is contained in a subset $S_i \in \mathcal{S}$. For each $S_i \in \mathcal{S}$, all maximal cliques of $G[S_i]$ can be enumerated in $O(3^{k/3})$ time by the algorithm in [42]. Therefore, all maximum cliques of G can be extracted in $O(3^{k/3}n)$ time. □

Lemma 7.6. *Suppose that G is a graph of bounded treewidth k with a nice tree decomposition $(T, \mathcal{S})$.*

(1) *The $\mathcal{VQ}$-incidence graph $\mathcal{H}'(G)$ is a partial $(k+1)$-tree.*

(2) *A tree decomposition of the $\mathcal{VQ}$-incidence graph $\mathcal{H}'(G)$ can be constructed in linear time.*

Proof. Clearly, $\omega(G) \le k + 1$ and G is also a partial $(k+1)$-tree. By Definition 4.1, $V(\mathcal{H}'(G)) = X(G) \cup S'(G)$, and each vertex in $S'(G)$ is adjacent to all vertices of exactly one maximum clique of G. It can be easily verified that there exists a $(k+1)$-tree H' such that $V(H') = V(G) \cup S'(G)$ and $\mathcal{H}'(G)$ is a subgraph of H'. Therefore, $\mathcal{H}'(G)$ is a partial $(k+1)$-tree. It was shown in [40] that for *each* constant k there is a linear-time algorithm for finding a treewidth decomposition for a graph of bounded treewidth k. Hence, a tree decomposition of $\mathcal{H}'(G)$ can be constructed in linear time. Following the discussion above, the lemma holds. □

Theorem 7.7. *The weighted maximum-clique transversal set problem can be solved in $O(4^k n)$ time for a graph G of bounded treewidth k with a tree decomposition $(T, \mathcal{S})$.*

Proof. Note that the weighted dominating set problem can be solved in $O(4^k n)$ time for a weighted graph $G = (V, E, w)$ of bounded treewidth k if a tree decomposition of G is given [43]. By Lemmas 4.3, 7.5, and 7.6, the theorem holds. □

8. Distance-Hereditary Graphs

This section deals with the weighted maximum-clique transversal set problem on distance-hereditary graphs. We show that the problem is linear-time solvable for distance-hereditary graphs.

A graph is *distance-hereditary* if any two distinct vertices have the same distance in every connected induced subgraph containing them. Chang et al. [44] showed that distance-hereditary graphs can be defined, recursively.

Theorem 8.1 (Chang et al. [44]). *Distance-hereditary graphs can be defined recursively as follows.*

(1) *A graph consisting of only one vertex is distance-hereditary, and the twin set is the vertex itself.*

(2) *If G_1 and G_2 are disjoint distance-hereditary graphs with the twin sets $TS(G_1)$ and $TS(G_2)$, respectively, then the graph $G = G_1 \cup G_2$ is a distance-hereditary graph and the twin set of G is $TS(G_1) \cup TS(G_2)$. G is said to be obtained from G_1 and G_2 by a false twin operation.*

(3) *If G_1 and G_2 are disjoint distance-hereditary graphs with the twin sets $TS(G_1)$ and $TS(G_2)$, respectively, then the graph G obtained by connecting every vertex of $TS(G_1)$ to all vertices of $TS(G_2)$ is a distance-hereditary graph, and the twin set of G is $TS(G_1) \cup TS(G_2)$. G is said to be obtained from G_1 and G_2 by a true twin operation.*

(4) *If G_1 and G_2 are disjoint distance-hereditary graphs with the twin sets $TS(G_1)$ and $TS(G_2)$, respectively, then the graph G obtained by connecting every vertex of $TS(G_1)$ to all vertices of $TS(G_2)$ is a distance-hereditary graph, and the twin set of G is $TS(G_1)$. G is said to be obtained from G_1 and G_2 by a pendant vertex operation.*

By Theorem 8.1, a distance-hereditary graph G has its own twin set $TS(G)$, the twin set $TS(G)$ is a subset of vertices of G, and it is defined recursively. The construction of G from disjoint distance-hereditary graphs G_1 and G_2 as described in Theorem 8.1 involves only the twin sets of G_1 and G_2.

Following Theorem 8.1, we can construct a binary ordered decomposition tree. In this decomposition tree, each leaf is a single vertex graph, and each internal node represents one of the three operations: pendant vertex operation (labeled by P), true twin operation (labeled by T), and false twin operation (labeled by F). This binary ordered decomposition tree is called a PTF-tree. It has $2n - 1$ tree nodes and can be obtained in linear time [44].

Definition 8.2. Let $G = (V, E)$ be a distance-hereditary graph and let $TS(G)$ be the twin set of G. We use $Q(G)$ to denote the collection of maximum cliques of G. Therefore $Q(G[TS(G)])$ is the collection of all maximum cliques of $G[TS(G)]$. Let $Q_E(G) = Q(G) \cup Q(G[TS(G)])$. The set $Q_E(G)$ denotes the collection of all maximum cliques of G and all maximum cliques of $G[TS(G)]$.

Definition 8.3. Suppose that G is a distance-hereditary graph obtained from two disjoint distance-hereditary graphs G_1 and G_2 by a true twin operation or a pendant vertex operation. We use $QX_{12}(G)$ to denote the set $\{Q_1 \cup Q_2 \mid Q_1 \in Q(G_1[TS(G_1)])$ and $Q_2 \in Q(G_2[TS(G_2)])\}$.

Lemma 8.4. *Let G be a distance-hereditary graph obtained from two disjoint distance-hereditary graphs G_1 and G_2 by a true twin operation or a pendant vertex operation. Then $QX_{12}(G) = Q(G[TS(G_1) \cup TS(G_2)])$.*

Proof. This lemma can be easily proved by contradiction. □

Lemma 8.5. *Suppose G is a graph obtained from two disjoint distance-hereditary graphs G_1 and G_2 by a true twin operation or a pendant vertex operation. Let $\mathcal{S}$ be a subset of $V(G)$ such that $\mathcal{S}$ intersects with all maximum cliques of $G[TS(G_1) \cup TS(G_2)]$. Then either $\mathcal{S} \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ or $\mathcal{S} \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$.*

Proof. Assume for contrary that neither $\mathcal{S} \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ nor $\mathcal{S} \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$. There exist maximum cliques Q_1 and Q_2 of $G_1[TS(G_1)]$ and $G_2[TS(G_2)]$, respectively, such that $\mathcal{S}$ does not contain any vertex in them. By Lemma 8.4, $Q_1 \cup Q_2$ is a maximum clique of $G[TS(G_1) \cup TS(G_2)]$. Then, $\mathcal{S}$ does not contain any vertex in $Q_1 \cup Q_2$, which contradicts the assumption that $\mathcal{S}$ intersects all maximum cliques of $G[TS(G_1) \cup TS(G_2)]$. Following the discussion above, the lemma holds. □

Definition 8.6. Let $G = (V, E, w)$ be a weighted graph. Suppose that $W = \{V_1, V_2, \dots, V_\ell\}$, where V_i is a subset of V for $1 \leq i \leq \ell$. Let $\min W$ be an element in W such that the weight of the element is minimum.

In the paper [1], Chang et al. presented a linear-time algorithm for the maximum-clique transversal set problem. Based upon their algorithm, we develop a linear-time algorithm to solve the weighted maximum-clique transversal set problem by the dynamic programming technique.

For a distance-hereditary graph $G = (V, E)$, we use $\omega(G)$ (resp., $\omega_t(G)$) to denote the clique number of G (resp., $G[TS(G)]$) and use $\mathcal{CT}(G)$ (resp., $\mathcal{CT}_t(G)$) to denote a maximum-clique transversal set of G (resp., $G[TS(G)]$) of minimum weight. A *strong maximum-clique transversal set* S of a distance-hereditary graph G is a subset of $V(G)$ such that S intersects all cliques in $Q_E(G)$. We use $\mathcal{SCT}(G)$ to denote a strong maximum-clique transversal set of G of minimum weight.

Lemma 8.7. *Suppose that G is a distance-hereditary graph of only one vertex v. Then, $\omega(G) = \omega_t(G) = 1$, and $\mathcal{CT}(G) = \mathcal{CT}_t(G) = \mathcal{SCT}(G) = \{v\}$.*

Proof. It follows from the definitions. □

Lemma 8.8. *Suppose that G is formed from two disjoint distance-hereditary graphs G_1 and G_2 by a "false twin" operation.*

(1) *If $\omega_t(G_1) = \omega_t(G_2)$ and $\omega(G_1) = \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}_t(G_2)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1) \cup \mathcal{SCT}(G_2)$.*

(2) *If $\omega_t(G_1) = \omega_t(G_2)$ and $\omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1)$, $\omega_t(G) = \omega_t(G_1) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}_t(G_2)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1) \cup \mathcal{CT}_t(G_2)$.*

(3) *If $\omega_t(G_1) = \omega_t(G_2)$ and $\omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_2) \cup \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_2) \cup \mathcal{CT}_t(G_1)$.*

(4) *If $\omega_t(G_1) > \omega_t(G_2)$ and $\omega(G_1) = \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}v(G_1) \cup \mathcal{CT}(G_2)$.*

(5) *If $\omega_t(G_2) > \omega_t(G_1)$ and $\omega(G_1) = \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_2)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_2) \cup \mathcal{CT}(G_1)$.*

(6) *If $\omega_t(G_1) > \omega_t(G_2)$ and $\omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1)$.*

(7) *If $\omega_t(G_2) > \omega_t(G_1)$ and $\omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_2)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_2)$.*

(8) *If $\omega_t(G_1) > \omega_t(G_2)$ and $\omega(G_1) < \omega(G_2)$, then $\omega(G) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)$.*

(9) *If $\omega_t(G_2) > \omega_t(G_1)$ and $\omega(G_2) < \omega(G_1)$, then $\omega(G) = \omega(G_1)$, $\omega_t(G) = \omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_2)$, and $\mathcal{SCT}(G) = \mathcal{CT}_t(G_2) \cup \mathcal{CT}(G_1)$.*

Proof. In the following, we just show the correctness for Statement (4) since other statements can be verified in similar ways. Note that $G = G_1 \cup G_2$ and $TS(G) = TS(G_1) \cup TS(G_2)$. Since $\omega_t(G_1) > \omega_t(G_2)$ and $\omega(G_1) = \omega(G_2)$, $\omega(G) = \omega(G_1) = \omega(G_2)$, and $\omega_t(G) = \omega_t(G_1)$. Therefore, $Q(G) = Q(G_1) \cup Q(G_2)$ and $Q(G[TS(G)]) = Q(G_1[TS(G_1)])$. We have $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$ and $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$.

By definition, $Q_E(G) = Q(G) \cup Q(G[TS(G)])$ and thus $Q_E(G) = Q(G_1) \cup Q(G_2) \cup Q(G_1[TS(G_1)])$. Then $Q_E(G) = Q_E(G_1) \cup Q(G_2)$. Hence, $\mathcal{SCT}(G) = \mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2)$. Following the discussion above, the statement is true. □

Lemma 8.9. *Suppose that G is formed from two disjoint distance-hereditary graphs G_1 and G_2 by a "pendant vertex" operation.*

(1) *If $\omega_t(G_1) + \omega_t(G_2) > \max\{\omega(G_1), \omega(G_2)\}$, then $\omega(G) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{CT}_t(G_1)$.*

(2) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_1), \mathcal{CT}_t(G_2) \cup \mathcal{CT}(G_1)\}$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1)$.*

(3) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_2), \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)\}$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)$.*

(4) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_1) = \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2), \mathcal{SCT}(G_2) \cup \mathcal{CT}(G_1)\}$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2)$.*

(5) *If $\omega_t(G_1) + \omega_t(G_2) < \omega(G_1)$ and $\omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1)$.*

(6) *If $\omega_t(G_1) + \omega_t(G_2) < \omega(G_2)$ and $\omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)$.*

(7) *If $\omega_t(G_1) + \omega_t(G_2) < \omega(G_2)$ and $\omega(G_2) = \omega(G_1)$, then $\omega(G) = \omega(G_1) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$, and $\mathcal{SCT}(G) = \mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2)$.*

Proof. In the following, we just show the correctness for Statement (3) since other statements can be verified in similar ways.

By Theorem 8.1, $TS(G) = TS(G_1)$ and G is obtained from G_1 and G_2 by connecting every vertex of $TS(G_1)$ to all vertices of $TS(G_2)$. Therefore, $\omega_t(G) = \omega_t(G_1)$ and $\mathcal{CT}_t(G) = \mathcal{CT}_t(G_1)$.

We now consider $\omega(G)$ and $\mathcal{CT}(G)$. In this case, $\omega_t(G_1) + \omega_t(G_2) = \omega(G_2) > \omega(G_1)$. Then, $\omega(G) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$ and thus $Q(G) = Q(G_2) \cup QX_{12}(G)$. Clearly, $\mathcal{CT}(G)$ is a subset of $TS(G_1) \cup V(G_2)$ and $\mathcal{CT}(G) \cap V(G_2)$ is a maximum-clique transversal set of G_2. By Lemma 8.4, we know that $QX_{12}(G) = G[TS(G_1) \cup TS(G_2)]$. Therefore, $\mathcal{CT}(G)$ intersects all maximum cliques of $G[TS(G_1) \cup TS(G_2)]$. By Lemma 8.5, either $\mathcal{CT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ or $\mathcal{CT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$. If $\mathcal{CT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$, then we have $\mathcal{CT}(G) = \mathcal{CT}(G_2) \cup \mathcal{CT}_t(G_1)$. If $\mathcal{CT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$, then $\mathcal{C}T(G) \cap V(G_2)$ is not only a maximum-clique transversal set of G but also a strong maximum-clique transversal set of G_2. We have $\mathcal{CT}(G) = \mathcal{SCT}(G_2)$. Hence, $\mathcal{CT}(G) = \min\{\mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2),\ \mathcal{SCT}(G_2)\}$.

We now consider $\mathcal{SCT}(G)$. By definition, $\mathcal{SCT}(G)$ intersects all maximum cliques in $Q(G[TS(G)])$ and $Q(G)$, respectively. Recall that $TS(G) = TS(G_1)$ and $Q(G) = Q(G_2) \cup QX_{12}(G)$. Therefore, $\mathcal{SCT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G[TS(G_1)]$ and it also intersects all maximum cliques in $QX_{12}(G)$. Besides, $\mathcal{SCT}(G) \cap V(G_2)$ is a maximum-clique transversal set of G_2. Hence, $\mathcal{SCT}(G) = \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)$.

Following the discussion above, the statement is true. □

Lemma 8.10. *Suppose that a distance-hereditary graph G is formed from two disjoint distance-hereditary graphs G_1 and G_2 by a "true twin" operation. Let $i \in \{1,2\}$.*

(1) *If $\omega_t(G_1) + \omega_t(G_2) > \max\{\omega(G_1), \omega(G_2)\}$, then $\omega(G) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1) + \omega_t(G_2)$, $\mathcal{CT}(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$.*

(2) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1) + \omega_t(G_2)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_1), \mathcal{CT}_t(G_2) \cup \mathcal{CT}(G_1)\}$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_1), \mathcal{CT}_t(G_2) \cup \mathcal{CT}(G_1)\}$.*

(3) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1) + \omega_t(G_2)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_2), \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)\}$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_2), \mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2)\}$.*

(4) *If $\omega_t(G_1) + \omega_t(G_2) = \omega(G_1) = \omega(G_2)$, then $\omega(G) = \omega(G_1) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$, $\omega_t(G) = \omega_t(G_1) + \omega_t(G_2)$, $\mathcal{CT}(G) = \min\{\mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2), \mathcal{SCT}(G_2) \cup \mathcal{CT}(G_1)\}$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2), \mathcal{SCT}(G_2) \cup \mathcal{CT}(G_1)\}$.*

(5) *If $\omega_t(G_1)+\omega_t(G_2) < \omega(G_1)$ and $\omega(G_1) > \omega(G_2)$, then $\omega(G) = \omega(G_1)$, $\omega_t(G) = \omega_t(G_1)+\omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1)$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_1), \mathcal{CT}(G_1) \cup \mathcal{CT}_t(G_2)\}$.*

(6) *If $\omega_t(G_1)+\omega_t(G_2) < \omega(G_2)$ and $\omega(G_2) > \omega(G_1)$, then $\omega(G) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)+\omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_2), \mathcal{CT}(G_2) \cup \mathcal{CT}_t(G_1)\}$.*

(7) *If $\omega_t(G_1)+\omega_t(G_2) < \omega(G_2)$ and $\omega(G_2) = \omega(G_1)$, then $\omega(G) = \omega(G_1) = \omega(G_2)$, $\omega_t(G) = \omega_t(G_1)+\omega_t(G_2)$, $\mathcal{CT}(G) = \mathcal{CT}(G_1) \cup \mathcal{CT}(G_2)$, $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$, and $\mathcal{SCT}(G) = \min\{\mathcal{SCT}(G_1) \cup \mathcal{CT}(G_2), \mathcal{SCT}(G_2) \cup \mathcal{CT}(G_1)\}$.*

Proof. In the following, we just show that the correctness for Statement (3) since other statements can be verified in similar ways.

By Theorem 8.1, $TS(G) = TS(G_1) \cup TS(G_2)$, and G is obtained from G_1 and G_2 by connecting every vertex in $TS(G_1)$ to all vertices in $TS(G_2)$. By Lemma 8.4, we have $Q(G[TS(G)]) = QX_{12}(G)$. By Lemma 8.5, either $\mathcal{CT}_t(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ or $\mathcal{CT}_t(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$. Hence, we have $\omega_t(G) = \omega_t(G_1)+\omega_t(G_2)$ and $\mathcal{CT}_t(G) = \min\{\mathcal{CT}_t(G_1), \mathcal{CT}_t(G_2)\}$.

We now consider $\omega(G)$ and $\mathcal{CT}(G)$. In this case, $\omega_t(G_1) + \omega_t(G_2) = \omega(G_2) > \omega(G_1)$. Then, $\omega(G) = \omega(G_2) = \omega_t(G_1) + \omega_t(G_2)$ and thus $Q(G) = Q(G_2) \cup QX_{12}(G)$. Clearly, $\mathcal{CT}(G)$ is a subset of $TS(G_1) \cup V(G_2)$ and $\mathcal{CT}(G) \cap V(G_2)$ is a maximum-clique transversal set of G_2. Note that $QX_{12}(G) = G[TS(G_1) \cup TS(G_2)] = G[TS(G)]$. By Lemma 8.5, either $\mathcal{CT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ or $\mathcal{CT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$. If $\mathcal{CT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$, then we have $\mathcal{CT}(G) = \mathcal{CT}(G_2) \cup \mathcal{CT}_t(G_1)$. If $\mathcal{CT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$, then $\mathcal{CT}(G) \cap V(G_2)$ is not only a maximum-clique transversal set of G but also a strong maximum-clique transversal set of G_2. We have $\mathcal{CT}(G) = \mathcal{SCT}(G_2)$. Hence, $\mathcal{CT}(G) = \min\{\mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2), \mathcal{SCT}(G_2)\}$.

We now consider $\mathcal{SCT}(G)$. By definition, $\mathcal{SCT}(G)$ intersects all maximum cliques in $Q(G[TS(G)])$ and $Q(G)$, respectively. Recall that $QX_{12}(G) = G[TS(G_1) \cup TS(G_2)] = G[TS(G)]$ and $Q(G) = Q(G_2) \cup QX_{12}(G)$. Clearly $\mathcal{SCT}(G) \cap V(G_2)$ is a maximum-clique transversal set of G_2. By Lemma 8.5, either $\mathcal{SCT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$ or $\mathcal{SCT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$. If $\mathcal{SCT}(G) \cap TS(G_1)$ is a maximum-clique transversal set of $G_1[TS(G_1)]$, then we have $\mathcal{SCT}(G) = \mathcal{CT}(G_2) \cup \mathcal{CT}_t(G_1)$. If $\mathcal{SCT}(G) \cap TS(G_2)$ is a maximum-clique transversal set of $G_2[TS(G_2)]$, then $\mathcal{SCT}(G) \cap V(G_2)$ is not only a strong maximum-clique transversal set of G but also a strong maximum-clique transversal set of G_2. We have $\mathcal{SCT}(G) = \mathcal{SCT}(G_2)$. Hence, $\mathcal{SCT}(G) = \min\{\mathcal{CT}_t(G_1) \cup \mathcal{CT}(G_2),\ \mathcal{SCT}(G_2)\}$.

Following the discussion above, the statement is true. □

We can develop a dynamic programming algorithm to solve the weighted maximum-clique transversal set problem in linear time for distance-hereditary graphs as follows.

Without loss of generality, we assume that G is a *connected* distance-hereditary graph with a vertex-weight function w. Given a PTF-tree PTF(G) of G rooted at node r, our algorithm starts from the leaves of PTF(G) and works upward to the root. For each node ℓ of PTF(G), let T_ℓ be the subtree of PTF(G) rooted at node ℓ, and let G_ℓ represent the subgraph of G induced by the leaves of T_ℓ. A node ℓ represents either a single-vertex subgraph of G or a subgraph G_ℓ of G obtained by applying one of pendant vertex, true twin, or false twin operations to children of node ℓ. We use ℓ_1 and ℓ_2 to denote the left and right children of node ℓ, respectively, if node ℓ is a nonleaf node of PTF(G). Therefore $G = G_r$ and G_r is formed from G_{r_1} and G_{r_2}. For each node ℓ, our algorithm computes $\omega(G_\ell)$, $\omega_t(G_\ell)$, $\mathcal{CT}(G_\ell)$, $\mathcal{CT}_t(G_\ell)$, and $\mathcal{SCT}(G_\ell)$, in $O(1)$ time by Lemmas 8.7–8.10. Then $\mathcal{CT}(G_r)$ is a maximum-clique transversal set of G of minimum weight. Note that PTF(G) has $2n - 1$ tree nodes and can be constructed in linear time. Hence, we obtain the following result.

Theorem 8.11. *If G is a weighted distance-hereditary graph, then maximum-clique transversal set of G of minimum weight can be computed in linear time.*

9. Conclusions

In this paper, we have presented polynomial-time algorithms (most of them with linear running time) for the weighted maximum-clique transversal set problem on split graphs, balanced graphs, strongly chordal graphs, Helly circular-arc graphs, comparability graphs, distance-hereditary graphs, and graphs of bounded treewidth. For further study, it is a great challenge to work on the complexity of this problem for other classes of graphs.

Acknowledgments

The author would like to thank the anonymous referees for their insightful comments which led to considerable improvements in the presentation of the paper. This research was partially supported by National Science Council in Taiwan, under the grant number NSC-97-2218-E-130-002-MY2.

References

[1] M.-S. Chang, T. Kloks, and C.-M. Lee, “Maximum clique transversals,” in *Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science*, vol. 2204 of *Lecture Notes in Computer Science*, pp. 32–43, 2001.

[2] C.-M. Lee, “Variations of maximum-clique transversal sets on graphs,” *Annals of Operations Research*, vol. 181, pp. 21–66, 2010.

[3] T. Andreae, M. Schughart, and Z. Tuza, “Clique-transversal sets of line graphs and complements of line graphs,” *Discrete Mathematics*, vol. 88, no. 1, pp. 11–20, 1991.

[4] T. Andreae and C. Flotow, “On covering all cliques of a chordal graph,” *Discrete Mathematics*, vol. 149, no. 1–3, pp. 299–302, 1996.

[5] T. Andreae, “On the clique-transversal number of chordal graphs,” *Discrete Mathematics*, vol. 191, no. 1–3, pp. 3–11, 1998.

[6] V. Balachandran, P. Nagavamsi, and C. P. Rangan, “Clique transversal and clique independence on comparability graphs,” *Information Processing Letters*, vol. 58, no. 4, pp. 181–184, 1996.

[7] H.-J. Bandelt and H. M. Mulder, “Distance-hereditary graphs,” *Journal of Combinatorial Theory. Series B*, vol. 41, no. 2, pp. 182–208, 1986.

[8] F. Bonomo, G. Durán, M. C. Lin, and J. L. Szwarcfiter, “On balanced graphs,” *Mathematical Programming*, vol. 105, no. 2-3, pp. 233–250, 2006.

[9] A. Brandstädt, V. D. Chepoi, and F. F. Dragan, “Clique r-domination and clique r-packing problems on dually chordal graphs,” *SIAM Journal on Discrete Mathematics*, vol. 10, no. 1, pp. 109–127, 1997.

[10] G. J. Chang, M. Farber, and Z. Tuza, “Algorithmic aspects of neighborhood numbers,” *SIAM Journal on Discrete Mathematics*, vol. 6, no. 1, pp. 24–29, 1993.

[11] M.-S. Chang, Y.-H. Chen, G. J. Chang, and J.-H. Yan, “Algorithmic aspects of the generalized clique-transversal problem on chordal graphs,” *Discrete Applied Mathematics*, vol. 66, no. 3, pp. 189–203, 1996.

[12] E. Dahlhaus, P. D. Manuel, and M. Miller, “Maximum h-colourable subgraph problem in balanced graphs,” *Information Processing Letters*, vol. 65, no. 6, pp. 301–303, 1998.

[13] G. Durán, M. C. Lin, S. Mera, and J. L. Szwarcfiter, “Algorithms for clique-independent sets on subclasses of circular-arc graphs,” *Discrete Applied Mathematics*, vol. 154, no. 13, pp. 1783–1790, 2006.

[14] G. Durán, M. C. Lin, S. Mera, and J. L. Szwarcfiter, “Algorithms for finding clique-transversals of graphs,” *Annals of Operations Research*, vol. 157, pp. 37–45, 2008.

[15] G. Durán, M. C. Lin, and J. L. Szwarcfiter, “On clique-transversals and clique-independent sets,” *Annals of Operations Research*, vol. 116, pp. 71–77, 2002.

[16] P. Erdős, T. Gallai, and Z. Tuza, “Covering the cliques of a graph with vertices,” *Discrete Mathematics*, vol. 108, no. 1-3, pp. 279–289, 1992.

[17] V. Guruswami and C. P. Rangan, “Algorithmic aspects of clique-transversal and clique-independent sets,” *Discrete Applied Mathematics*, vol. 100, no. 3, pp. 183–202, 2000.

[18] C.-M. Lee and M.-S. Chang, “Distance-hereditary graphs are clique-perfect,” *Discrete Applied Mathematics*, vol. 154, no. 3, pp. 525–536, 2006.

[19] E. Shan, Z. Liang, and T. C. E. Cheng, “Clique-transversal number in cubic graphs,” *Discrete Mathematics & Theoretical Computer Science. DMTCS.*, vol. 10, no. 2, pp. 115–124, 2008.

[20] G. Xu, E. Shan, L. Kang, and T. C. E. Cheng, “The algorithmic complexity of the minus clique-transversal problem,” *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1410–1418, 2007.

[21] A. Brandstädt, V. B. Le, and J. P. Spinrad, “Graph classes—a Survey,” *SIAM Monographs on Discrete Mathematics and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1999.

[22] D. R. Fulkerson, A. J. Hoffman, and R. Oppenheim, “On balanced matrices,” *Mathematical Programming Study*, no. 1, pp. 120–132, 1974.

[23] E. Prisner, “Hereditary clique-Helly graphs,” *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 14, pp. 216–220, 1993.

[24] K. Makino and T. Uno, “New algorithms for enumerating all maximal cliques,” in *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory*, vol. 3111 of *Lecture Notes in Computer Science*, pp. 260–274, 2004.

[25] D. J. Rose, “Triangulated graphs and the elimination process,” *Journal of Mathematical Analysis and Applications*, vol. 32, pp. 597–609, 1970.

[26] M. Farber, “Characterizations of strongly chordal graphs,” *Discrete Mathematics*, vol. 43, no. 2-3, pp. 173–189, 1983.

[27] R. Paige and R. E. Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.

[28] J. P. Spinrad, "Doubly lexical ordering of dense 0–1 matrices," *Information Processing Letters*, vol. 45, no. 5, pp. 229–235, 1993.

[29] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 180–187, 1972.

[30] M. Farber, "Domination, independent domination, and duality in strongly chordal graphs," *Discrete Applied Mathematics*, vol. 7, no. 2, pp. 115–130, 1984.

[31] M. C. Lin, R. M. McConnell, F. J. Soulignac, and J. L. Szwarcfiter, "On cliques of Helly circular-arc graphs," *Electronic Notes in Discrete Mathematices*, vol. 30, pp. 117–122, 2008.

[32] M.-S. Chang, "Efficient algorithms for the domination problems on interval and circular-arc graphs," *SIAM Journal on Computing*, vol. 27, no. 6, pp. 1671–1694, 1998.

[33] R. M. McConnell and J. P. Spinrad, "Modular decomposition and transitive orientation," *Discrete Mathematics*, vol. 201, no. 1–3, pp. 189–241, 1999.

[34] L. Rédei, "Ein Kombinatorischer Satz," *Acta Litteraria Szeged*, vol. 7, pp. 39–43, 1934.

[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 3rd edition, 2009.

[36] L. R. Ford, Jr. and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[37] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 1962.

[38] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the Association for Computing Machinery*, vol. 35, no. 4, pp. 921–940, 1988.

[39] T. Kloks, *Treewidth—Computations and Approximation*, vol. 842 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1994.

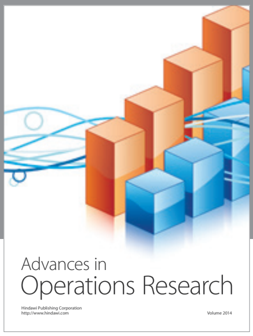
[40] H. Bodlaender, "A linear-time algorithm for finding tree-decompositions of small treewidth," *SIAM Journal on Computing*, vol. 25, no. 6, pp. 1305–1317, 1996.

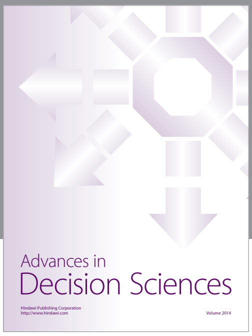
[41] H. Bodlaender and R. H. Möhring, "The pathwidth and treewidth of cographs," *SIAM Journal on Discrete Mathematics*, vol. 6, no. 2, pp. 181–188, 1993.

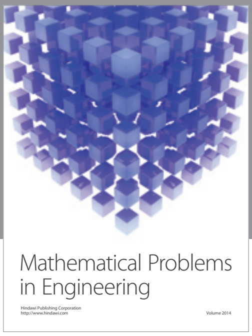
[42] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.

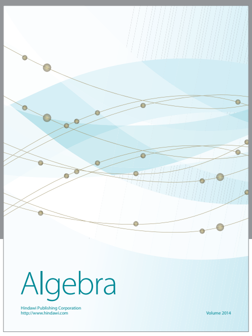
[43] J. Alber and R. Niedermeier, "Improved tree decomposition based algorithms for domination-like problems," in *Proceedings of the LATIN*, vol. 2286 of *Lecture Notes in Computer Science*, pp. 613–627, Springer, Berlin, Germany, 2002.

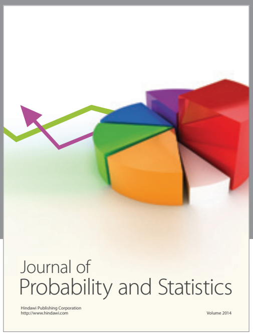
[44] M.-S. Chang, S.-y. Hsieh, and G.-H. Chen, "Dynamic programming on distance-hereditary graphs," in *Proceedings of the 8th International Syposium on Algorithms and Computation*, vol. 1350 of *Lecture Notes in Computer Science*, pp. 344–353, 1997.


Advances in
Operations Research

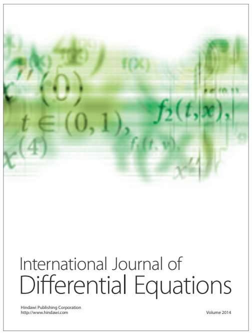

Advances in
Decision Sciences

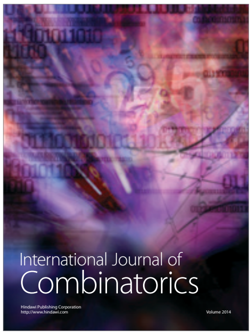

Mathematical Problems
in Engineering

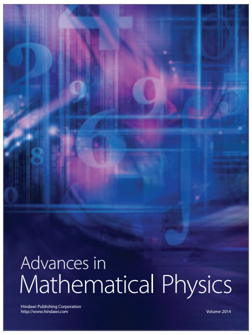

Algebra

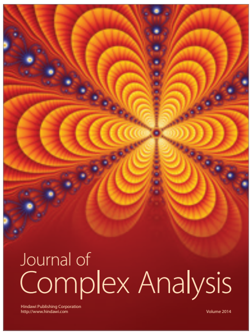

Journal of
Probability and Statistics

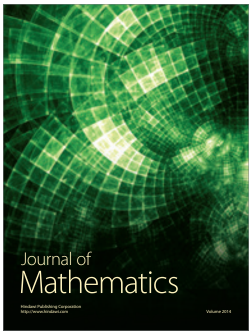
The Scientific
World Journal

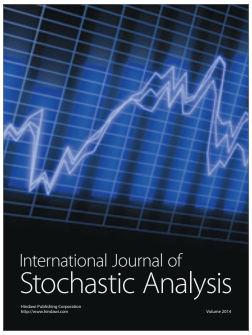

International Journal of
Differential Equations

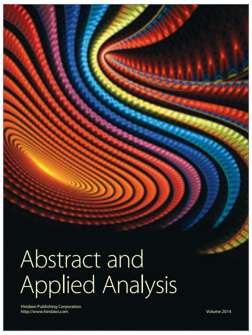

International Journal of
Combinatorics

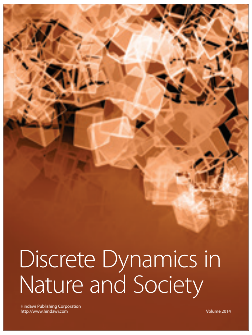

Advances in
Mathematical Physics

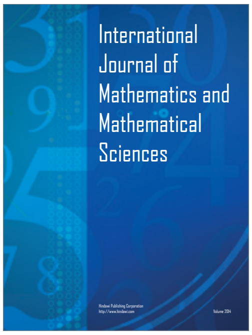

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

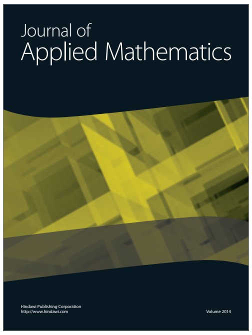

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization