symmetry

Article

Remarks on Parameterized Complexity of Variations of the Maximum-Clique Transversal Problem on Graphs

Chuan-Min Lee

Department of Computer and Communication Engineering, Ming Chuan University, 5 De Ming Road, Guishan District, Taoyuan City 333, Taiwan; joneslee@mail.mcu.edu.tw; Tel.: +886-3-350-7001 (ext. 3432); Fax: +886-3-359-3876

Abstract: With the rapid growth in the penetration rate of mobile devices and the surge in demand for mobile data services, small cells and mobile backhaul networks have become the critical focus of next-generation mobile network development. Backhaul requirements within current wireless networks are almost asymmetrical, with most traffic flowing from the core to the handset, but 5G networks will require more symmetrical backhaul capability. The deployment of small cells and the placement of transceivers for cellular phones are crucial in trading off the symmetric backhaul capability and cost-effectiveness. The deployment of small cells is related to the placement of transceivers for cellular phones. Chang, Kloks, and Lee transformed the placement problem into the maximum-clique transversal problem on graphs. From the theoretical point of view, our paper considers the parameterized complexity of variations of the maximum-clique transversal problem for split graphs, doubly chordal graphs, planar graphs, and graphs of bounded treewidth.

Keywords: parameterized complexity; asymmetric networks; signed maximum-clique transversal function; minus maximum-clique transversal function; symmetrical backhaul capability

check for updates

Citation: Lee, C.-M. Remarks on Parameterized Complexity of Variations of the Maximum-Clique Transversal Problem on Graphs. *Symmetry* **2022**, *14*, 676. https://doi.org/10.3390/sym14040676

Academic Editor: Manuel Lafond

Received: 6 March 2022
Accepted: 23 March 2022
Published: 24 March 2022

1. Introduction

Fixed/mobile network convergence enhances the competitive advantage of telecommunications operators/companies. With the rapid growth in the penetration rate of mobile devices and the surge in demand for mobile data services, telecommunications operators are stepping up their pace to actively improve their wireless network infrastructure to cope with the advent of the mobile broadband networks era. They have to extend broadband networks to any place through the seamless connection between fixed and mobile networks, which helps accelerate the deployment time, reduces maintenance costs, and further enhances market competitiveness. Under the circumstances, small cells and mobile backhaul networks have become the critical focus of next-generation mobile network development.

The so-called mobile backhaul network transmits the mobile signal traffic between the base station and the mobile terminal device to the wireless node, and then aggregates and transmits it to the telecommunications core network. Backhaul requirements within current wireless networks are almost asymmetrical, with most traffic flowing from the core to the handset, but 5G networks will require more symmetrical backhaul capability. Furthermore, we must backhaul massive broadband traffic from small cells to the control center when small cells grow very large. Hence, the deployment of small cells and the placement of transceivers for cellular phones are crucial in trading off the symmetric backhaul capability and cost-effectiveness. The deployment of small cells is related to the placement of transceivers for cellular phones. Chang, Kloks, and Lee transformed the placement problem into the maximum-clique transversal problem on graphs [1]. One of their main objectives is as follows. Modern cellular telecommunications systems divide the entire service area into a set of small regions, which are called cells. Cells are generally thought of as hexagonal grids. One standard method used to place transceivers for cellular telephones is to place them at the corner points of each hexagonal grid. Since the need for communication proliferates, one transceiver cannot handle all communication requirements in its reach. The most widely used solution is to place another transceiver close to it. The system needs to assign a different frequency to this new transceiver to avoid interference. However, the number of available frequencies is limited. The rapidly increasing demand for communications makes the number of transceivers, placed close to each other, grow very large. It is desirable to replace them with a more efficient big transceiver tower. These towers are pretty large and rise hundreds of feet into the air. They contain very costly hardware to switch between different frequencies between transceivers in some optimal way to allocate and handle them with great care. Nevertheless, since transceiver towers are very costly, their number is expected to be as small as possible. The placement of transceivers for cellular telephones motivated Chang et al. to introduce the maximum-clique transversal problem on graphs [1]. Later, several studies have proposed and worked on variations of the maximum-clique transversal set problem from the theoretical point of view [2–9]. Most of them either develop algorithms to solve the problems and evaluate algorithms' performance by *asymptotic analysis* (based on big O notation) to see if the algorithms can solve the problems in *polynomial time* or prove the *NP-completeness* of the problems for some graph classes. Their approaches are concerned with two classical computational complexity classes: P and NP.

This paper considers the parameterized complexity of the clique transversal problem (CTP) and variations of the maximum-clique transversal problem (MCTP) on graphs, such as the k-fold maximum-clique transversal problem (k-FMCTP), the $\{k\}$-maximum-clique transversal problem ($\{k\}$-MCTP), the signed maximum-clique trasversal problem (SMCTP), and the minus maximum-clique transversal problem (MMCTP).

Parameterized complexity is a new branch of computational complexity theory. Consider an algorithm for a parameterized problem (I, k), where I is the problem instance and k the parameter. The algorithm is uniformly polynomial if it runs in $O(f(k)|I|^c)$ time, where $|I|$ is the size of I, $f(k)$ an arbitrary function, and c a constant independent of k. A parameterized problem is fixed-parameter tractable (FPT) if it admits a uniformly polynomial algorithm [10]. An fpt-reduction is a reduction transforming an instance of some parameterized problem into an equivalent instance of another parameterized problem and can be computed in uniformly polynomial time. A parameterized problem is para-NP-complete if it is NP-complete for fixed values of the parameter(s).

The W-hierarchy is another way to classify parameterized problems into computation complexity classes [10]. A parameterized problem is in the class $W[i]$ if it is fpt-reducible to the circuit-satisfiability problem that has weft at most i [10]. If a problem $Q \in W[i]$ and every problem in $W[i]$ can be fpt-reduced to Q, problem Q is $W[i]$-complete. Note that $FPT = W[0]$ and $W[i] \subseteq W[j]$ for all $i < j$.

There are very few algorithmic results for the parameterized complexity of the CTP and variations of the MCTP. Table 1 lists previous results and our fixed parameter **intractable** results for the considered problems, and uses "$W[2]$-c" and "para-NP-c" to represent "$W[2]$-complete" and "para-NP-complete", respectively. Table 2 lists previous results and our fixed parameter **tractable** results for the considered problems and uses n and ω to represent the number of vertices and the maximum size of a clique in a graph. Both tables use starred entries to denote our results. In Table 2, c is a moderate constant. As far as we know, $c = 4^{6\sqrt{34}}$ by the paper [11]. The rest of the paper is organized as follows.

1. Section 2 reviews the definitions of the considered problems and the most well-known notions from graph theory.
2. We prove in Section 3 that the clique transversal problem parameterized by the solution size is $W[2]$-complete for split graphs, and the following problems are para-NP-complete: the minus maximum-clique transversal problem parameterized by the solution weight for planar graphs, and the signed maximum-clique transversal problem parameterized by the solution weight for doubly chordal graph and planar graphs with clique number three.

3. We show the FPT results for graphs of bounded treewidth in Section 4.
   Section 4.1 shows that the k-fold maximum-clique transversal problem can be solved in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time for any graph G with bounded treewidth $tw(G)$.
   Section 4.2 reduces the $\{k\}$-maximum-clique transversal problem to the k-fold maximum-clique transversal problem and solves the problem in $O(2^{k(tw(G)+1)} \cdot k^2 \cdot (tw(G)+1) \cdot |V(G)|)$ time. We develop a dynamic programming algorithm to improve the complexity of the problem to $O((k+1)^{tw(G)+1} \cdot tw(G) \cdot |V(G)|)$ time.
   Section 4.3 deals with the signed and minus maximum-clique transversal problems. We reduce the signed and minus maximum-clique transversal problems to the k-fold maximum-clique transversal problem and solve these problems in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ and $O(4^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time, respectively. The complexity of the minus maximum-clique transversal problem for graphs of bounded treewidth can be improved to $O(3^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ by our dynamic programming technique used for the $\{k\}$-maximum-clique transversal problem.
4. Finally, we conclude the paper and present some future works in Section 5.

Table 1. Previous results and our fixed parameter intractable results.

Parameter: The Solution Size or Weight					
Graphs	**CTP**	**k-FMCTP**	**$\{k\}$-MCTP**	**SMCTP**	**MMCTP**
Chordal	$W[2]$-c *	$W[2]$-c [1] ($k=1$)	$W[2]$-c [1] ($k=1$)	para-NP-c *	–
Doubly Chordal	–	–	–	para-NP-c *	–
Split	$W[2]$-c *	–	–	–	–
Planar	–	–	–	para-NP-c *	para-NP-c *

Table 2. Previous results and our fixed parameter tractable results.

Problem	**Bounded Treewidth Graph Parameter: Treewidth t**	**Planar Graph Parameter: The Solution Size ℓ**
CTP	$O(4^t \cdot t \cdot n)$ [6]	*–
MCTP	$O(2^t \cdot t \cdot n)^*$	$\min\{O(\omega^\ell \cdot \ell^2+n), O(c^{\sqrt{\ell}} \cdot \ell^2+n)\}$ [1]
k-FMCTP	$O(2^t \cdot t \cdot n)^*$	–
$\{k\}$-MCTP	$O((k+1)^{t+1} \cdot t \cdot n)^*$	–
SMCTP	$O(2^t \cdot t \cdot n)^*$	–
MMCTP	$O(3^t \cdot t \cdot n)^*$	–

2. Definitions and Notations

All graphs in this paper are undirected graphs without self-loops and multiple edges. Let $G=(V,E)$ be a graph with $|V|=n$ and $|E|=m$. The vertex set of G is also referred to as $V(G)$ and its edge set as $E(G)$. We use uv to denote the edge between the vertices u and v in a graph. If any two distinct vertices in a graph are adjacent, then the graph is a complete graph. We use $G[W]$ to denote the subgraph of G induced by a subset W of V. The neighborhood of a vertex v in a graph G, denoted by $N_G(v)$, is the set of all v's neighbors in G. The closed neighborhood of a vertex v in G, denoted by $N_G[v]$, is $N_G(v) \cup \{v\}$. The degree of a vertex v in G, denoted by $deg_G(v)$, is the number of v's neighbors in G. A dominating set of G is a set $D \subseteq V$ such that $|D \cap N_G[v]| \geq 1$ for every $v \in V$. The domination number of G, denoted by $\gamma(G)$, is the minimum cardinality of a dominating set of G. The domination problem is to find a minimum dominating set of G.

A set $S \subseteq V$ is a clique if $xy \in E$ for any two distinct vertices $x, y \in S$. If $|S|=k$, then S is a k-clique. If a clique C is not a proper subset of any other clique, then C is a *maximal* clique. A set $D \subseteq V$ is a *clique transversal set* of G if $|C \cap D| \geq 1$ for every maximal clique C 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 transversal problem is to find a clique transversal set of G of minimum cardinality.

A maximal clique S is a maximum clique if $|S| \geq |S'|$ for any maximal clique S' of G. 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 set of all maximum cliques of G. A set $D \subseteq V$ is a maximum-clique transversal set of G if $|D \cap Q| \geq 1$ for every $Q \in Q(G)$. 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 problem is to find a maximum-clique transversal set of G of minimum cardinality.

Assume that $Y \subset \mathbb{R}$ and $f : S \to Y$ is a function. Let $f(S') = \sum_{s \in S'} f(s)$ for $S' \subseteq S$, and let $f(S)$ be the weight of f. A maximum-clique transversal set of G can be expressed as a function f whose domain is $V(G)$ and range is $\{0, 1\}$, and $f(Q) \geq 1$ for $Q \in Q(G)$. The function f is a maximum-clique transversal function of G and $\tau_M(G)$ is the minimum weight of a maximum-clique transversal function of G. The k-fold maximum-clique, $\{k\}$-maximum-clique, minus maximum-clique, and signed maximum-clique transversal problems are variations of the maximum-clique transversal set problem. They are defined as follows.

Definition 1 ([3])**.** *Suppose that $k \in \mathbb{Z}^+$ is fixed and G is a graph. A set $D \subseteq V(G)$ is a k-fold maximum-clique transversal set of G if $|Q \cap D| \geq k$ for $Q \in Q(G)$. The number $\tau_{\times k}(G) = min\{|S| \mid S$ is a k-fold maximum-clique transversal set of $G\}$ is the k-fold maximum-clique transversal number of G. The k-fold maximum-clique transversal problem is to find a k-fold maximum-clique transversal set of G of minimum cardinality.*

Definition 2 ([3])**.** *Suppose that $k \in \mathbb{Z}^+$ is fixed and G is a graph. A function f is a $\{k\}$-maximum-clique transversal function of G if the domain and range of f are $V(G)$ and $\{0, 1, 2, \ldots, k\}$, respectively, and $f(Q) \geq k$ for $Q \in Q(G)$. The number $\tau_{\{k\}}(G) = min\{f(V(G)) \mid f$ is a $\{k\}$-maximum-clique transversal function of $G\}$ is the $\{k\}$-maximum-clique transversal number of G. The $\{k\}$-maximum-clique transversal problem is to find a $\{k\}$-maximum-clique transversal function of G of minimum weight.*

Definition 3 ([3])**.** *Suppose that G is a graph. A function f is a signed maximum-clique transversal function of G if the domain and range of f are $V(G)$ and $\{-1, 1\}$, respectively, and $f(Q) \geq 1$ for $Q \in Q(G)$. If the domain and range of f are $V(G)$ and $\{-1, 0, 1\}$, respectively, and $f(Q) \geq 1$ for $Q \in Q(G)$, then f is a minus maximum-clique transversal function of G. The number $\tau_M^s(G) = min\{f(V(G)) \mid f$ is a signed maximum-clique transversal function of $G\}$ is the signed maximum-clique transversal number of G. The minus maximum-clique transversal number of G is $\tau_M^-(G) = min\{f(V(G)) \mid f$ is a minus maximum-clique transversal function of $G\}$. The minus (signed) maximum-clique transversal problem is to find a minus (signed) maximum-clique transversal function of G of minimum weight.*

3. The Fixed-Parameter Intractable Results

A chord of a cycle is an edge joining two non-consecutive vertices of the cycle. A graph is chordal if it does not contain a chordless cycle of length greater than three [12]. A graph is split if its vertices can be partitioned into two sets C and S, such that C is a clique and S is an independent set.

Theorem 1. *The clique transversal problem parameterized by the solution size is $W[2]$-complete for split graphs.*

Proof. Assume that $G = (V, E)$ is a graph with $V = \{v_i \mid 1 \leq i \leq n\}$. Let $X = \{x_i \mid 1 \leq i \leq n\}$ and let H be a split graph obtained from G by $V(H) = V \cup X$ and $E(H) = \{v_iv_j \mid 1 \leq i < j \leq n\} \cup \{x_iv | v \in N_G[v_i], 1 \leq i \leq n\}$. In H, the maximal cliques are the sets V and $N_G[v_i] \cup \{x_i\}$ for $1 \leq i \leq n$. A minimum dominating set of G is a clique transversal set of H. Clearly, there exists a minimum clique transversal set D of H such that

$D \cap X = \emptyset$. Then, a minimum clique transversal set of H is a dominating set of G. We have $\gamma(G) = \tau_C(H)$ following the discussion above. Hence, $\tau_C(H) \leq k$ if, and only if, $\gamma(G) \leq k$. The domination problem for a graph G can be reduced to the clique transversal problem for a split graph H. Since the domination problem parameterized by the solution size k is $W[2]$-complete [10], the clique transversal problem parameterized by the solution size is $W[2]$-complete for split graphs. □

Remark 1. *The maximum-clique transversal is a particular case of the k-fold maximum-clique transversal problem with $k = 1$. The k-fold maximum-clique, $\{k\}$-maximum-clique, minus maximum-clique, and signed maximum-clique transversal problems are linear-time solvable for split graphs [3].*

Consider the following two decision problems.

1. **The nonpositive minus maximum-clique transversal problem.**
 Instance: A graph G
 Question: Does G have a minus maximum-clique transversal function of weight at most 0?
2. **The nonpositive signed maximum-clique transversal problem.**
 Instance: A graph G
 Question: Does G have a signed maximum-clique transversal function of weight at most 0?

Definition 4. *Let k be a nonnegative integer and let F be a set of k vertices. The f_k-transformation of a graph $G = (V, E)$, denoted by $f_k(G)$, is the graph $H = (V \cup F, E)$.*

Theorem 2. *Let $\mathcal{G}$ and $\mathcal{H}$ be graph classes such that $\mathcal{H} = \{f_k(G) \mid G \in \mathcal{G}\}$. The following statements are true.*

(1) If the signed (minus) maximum-clique transversal problem is NP-complete for $\mathcal{G}$, the nonpositive signed (minus) maximum-clique transversal problem is NP-complete for $\mathcal{H}$.

(2) If the nonpositve signed (minus) maximum-clique transversal problem is NP-complete for graph class $\mathcal{H}$, the signed (minus) maximum-clique transversal problem parameterized by the solution weight is para-NP-complete for $\mathcal{H}$.

Proof. (1) Let k be a nonnegative integer and let F be a set of k vertices. Let $G = (V, E) \in \mathcal{G}$ and $H = (V \cup F, E) \in \mathcal{H}$. In H, every vertex $x \in F$ is isolated and not in any maximum clique of H. Then, $f(x) = -1$ for any signed (minus) maximum-clique transversal function f of H of "minimum weight". Clearly, $\tau^s_M(H) = \tau^s_M(G) - |F| = \tau^s_M(G) - k$. It follows that $\tau^s_M(H) \leq 0$ if and only if $\tau^s_M(G) \leq k$. Similarly, $\tau^-_M(H) = \tau^-_M(G) - |F| = \tau^-_M(G) - k$. Then, $\tau^-_M(H) \leq 0$ if and only if $\tau^-_M(G) \leq k$. The statement (1) is therefore true.
(2) The nonpositive signed (minus) maximum-clique transversal problem for $\mathcal{H}$ is a particular case of the signed (minus) maximum-clique transversal problem parameterized by solution weight for $\mathcal{H}$. Hence, the statement (2) is true. □

Corollary 1. *Let $\mathcal{G}_1$ be the graph class of planar graphs and let $\mathcal{G}_2$ be the graph class of doubly chordal graphs and planar graphs with the clique number 3. The following statements are true.*

(1) The nonpositive minus maximum-clique transversal problem is NP-complete for $\mathcal{G}_1$.

(2) The nonpositive signed maximum-clique transversal problem is NP-complete for $\mathcal{G}_2$.

Proof. (1) Let $G \in \mathcal{G}_1$ and $H = f_k(G)$. Clearly, $H \in \mathcal{G}_1$. It is known that the minus maximum-clique transversal problem is NP-complete for $\mathcal{G}_1$ [3]. By Theorem 2, the nonpositive minus maximum-clique transversal problem is NP-complete for $\mathcal{G}_1$.
(2) Let $G \in \mathcal{G}_2$ and $H = f_k(G)$. Clearly, $H \in \mathcal{G}_2$. It is known that the signed maximum-clique transversal problem is NP-complete for $\mathcal{G}_2$ [3]. By Theorem 2, the nonpositive signed maximum-clique transversal problem is NP-complete for $\mathcal{G}_2$. □

Corollary 2. *Let $\mathcal{G}_1$ be the graph class of planar graphs and let $\mathcal{G}_2$ be the graph class of doubly chordal graphs and planar graphs with the clique number 3. The following statements are true.*

(1) The minus maximum-clique transversal problem parameterized by the solution weight is para-NP-complete for $\mathcal{G}_1$.
(2) The signed maximum-clique transversal problem parameterized by the solution weight is para-NP-complete for $\mathcal{G}_2$.

Proof. The corollary holds by Theorem 2 and Corollary 1. □

4. Fixed-Parameter Tractable Results for Graphs of Bounded Treewidth

Assume that G is a graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$. Let $N_i[v]$ denote the closed neighborhood of v in $G[\{v_i, v_{i+1}, \ldots, v_n\}]$. The ordering $(v_1, v_2, \ldots, v_n)$ of V is a *perfect elimination ordering* of G if for all $i \in \{1, \ldots, n\}$, $N_i[v_i]$ is a clique. A graph G is chordal if and only if G has a perfect elimination ordering [13].

A chordal graph H with n vertices is a t-tree if and only if either H is a complete graph of t vertices or H has more than t vertices and there exists a perfect elimination ordering $(v_1, v_2, \ldots, v_n)$ such that $N_i[v_i]$ is a $(t+1)$-clique for $1 \leq i \leq n-t$. For convenience, we define a t-tree to have at least $t+1$ vertices. With this definition, a complete graph of t vertices is a $(t-1)$-tree.

A triangulation of a graph G is a graph H with the same set of vertices such that H is chordal and G is a subgraph of H. We say that G is triangulated into H. Subgraphs of t-trees are called *partial t-trees*. The *treewidth* of a graph G is the minimum value t such that G is a partial t-tree. It is clear that a graph of treewidth t is also a partial ℓ-tree for every $\ell \geq t$. The class of partial t-trees is exactly the class of graphs of treewidth at most t. Therefore, $\omega(G) = t+1$ for any graph G of treewidth t. The treewidth of a graph can be defined by the concept of *tree decomposition* of a graph.

Definition 5. *A tree decomposition of a graph G is a pair $D = (T, B)$, where T is a tree with $V(T) = \{1, 2, \ldots, \ell\}$ and $B = \{B_i \mid i \in V(T)\}$ is a collection of subsets of vertices of G such that the following three conditions are satisfied.*

1. *Every vertex $x \in V(G)$ appears in at least one set $B_i \in B$.*
2. *For every edge $e \in E(G)$, there is at least one set $B_i \in B$ containing both vertices of e.*
3. *For each vertex $x \in V(G)$, the set $\{i \mid x \in B_i\}$ forms a subtree of T.*

For simplicity, we use T to represent the vertex set of T if T is understood as a tree and we call its vertices *nodes*. We also refer to all elements of B as *bags*. Lemma 1 shows an alternative way to formulate the third condition of Definition 5.

Lemma 1. *Let $D = (T, \{B_i \mid i \in T\})$ be a tree decomposition of a graph G. If a vertex x appears in two bags $B_i, B_j \in B$, then it appears in every bag B_k for the node k on the tree path from node i to node j in T.*

Definition 6. *The width of a tree decomposition $D = (T, \{B_i \mid i \in T\})$ of a graph G is $\max_{i \in T}\{|B_i|\} - 1$. The treewidth of a graph G, denoted by $tw(G)$, is the minimum width over all tree decompositions of G.*

Theorem 3 (See also Lemma 2.1.13 in [14])**.** *If G is a chordal graph with at least $t+1$ vertices and $\omega(G) \leq t+1$, then G can be triangulated into a t-tree.*

Theorem 4 ([15])**.** *Any partial t-tree with at least $t+1$ vertices can be triangulated into a t-tree with the same number of vertices.*

Definition 7. *A tree decomposition is rooted if the tree is equipped with a root node. A nice tree decomposition $D = (T, \{B_i \mid i \in T\})$ is a rooted tree decomposition satisfying the following conditions.*

1. *Every node of T has at most two child nodes.*
2. *If a node i has two child nodes—j and k—then $B_i = B_j = B_k$. The node is called a join node.*
3. *If a node i has only one child node j, then either (1) $B_j \subset B_i$ and $|B_i| = |B_j| + 1$, or (2) $B_i \subset B_j$ and $|B_i| = |B_j| - 1$. In the case (1), the node is called an introduce node, whereas in the case (2), the node is called a forget node.*

Remark 2. *Figure 1 shows a nice tree decomposition $D = (T, \{B_i \mid i \in T\})$ of a graph G, where T has 17 nodes. Node 3 is a join node since it has two child nodes, 4 and 9, and $B_3 = B_4 = B_9 = \{b, c, d\}$. Node 6 is an introduce node since it has only one child node 7, and $B_7 \subset B_6$ and $|B_6| = |B_7| + 1$. Node 10 is a forget node since it has only one child node 11, and $B_{10} \subset B_{11}$ and $|B_{10}| = |B_{11}| - 1$.*

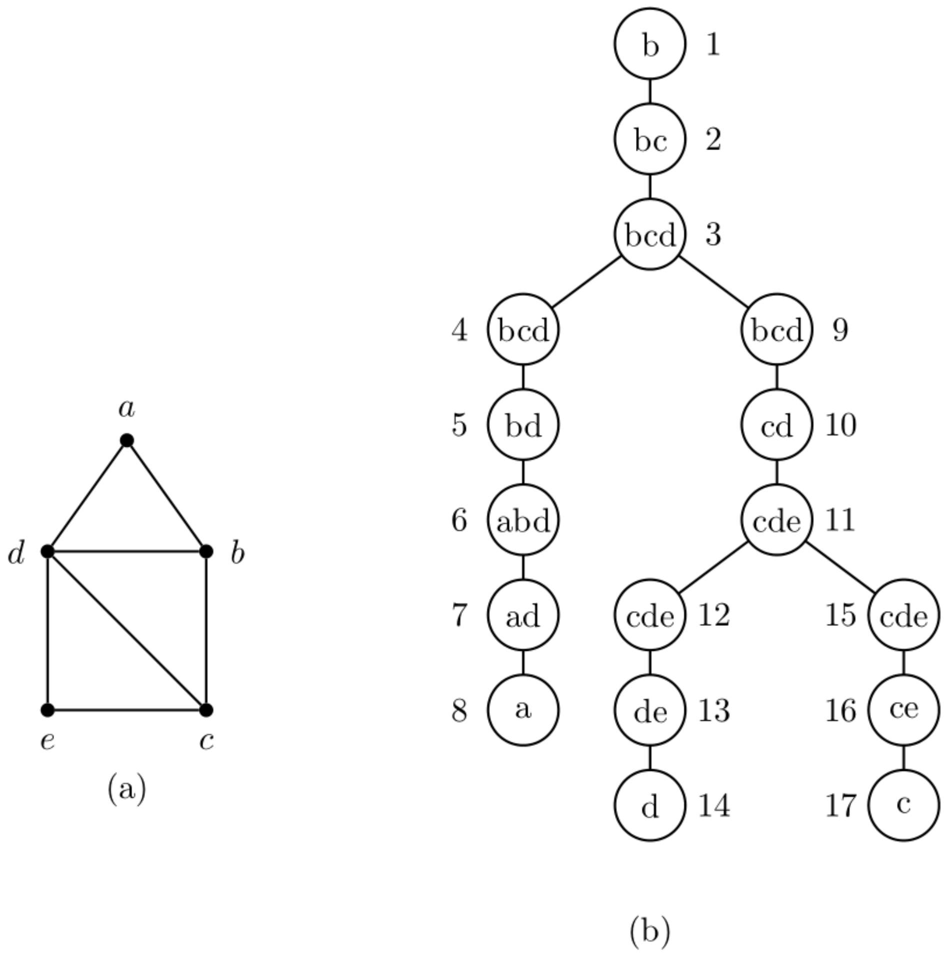

Figure 1. (**a**) A graph G. (**b**) A nice tree decomposition of G.

Lemma 2 ([14]). *For any constant t, given a tree decomposition of a graph G of width t and $O(|V(G)|)$ nodes, one can find a nice tree decomposition of G of width t and with at most $4 \cdot |V(G)|$ nodes in $O(|V(G)|)$ time.*

This paper assumes that a nice tree decomposition of a graph G of width $tw(G)$ and with $O(|V(G)|)$ nodes is part of the input by Lemma 2.

4.1. The k-Fold Maximum-Clique Transversal Problem

This section studies the k-fold maximum-clique transversal problem on graphs of bounded treewidth.

Definition 8. *Let $D = (T, \{B_i \mid i \in T\})$ be a rooted tree decomposition of G. For each node $i \in T$, let T_i be the subtree of T rooted at i and and let G_i be the induced subgraph $G[\cup_{s \in T_i} B_s]$. Clearly, $T = T_r$ and $G = G_r$.*

Theorem 5 ([16])**.** *Assume that $D = (T, B)$ is a tree decomposition of a graph G. For every clique C of G, there exists a bag $B_p \in B$ such that $C \subseteq B_p$.*

Lemma 3. *Let $D = (T, B)$ be a tree decomposition of a graph G of width $tw(G)$. For each maximum clique Q of G, there exists a bag $B_p \in B$ such that $Q = B_p$.*

Proof. Following Theorem 5, there exists a bag $B_p \in B$ such that $Q \subseteq B_p$ for a maximum clique Q of G. Then, $|Q| \leq |B_p|$. Clearly, $|B_p| \leq tw(G) + 1 \leq \omega(G) = |Q|$. Hence, $Q = B_p$. □

Definition 9. *Let G be a graph of bounded treewidth with a nice tree decomposition $(T, \{B_i \mid i \in T\})$ rooted at node r. Let k be a fixed positive integer such that $k \leq \omega(G)$. For each node $i \in T$, let $F_{k,i}(X)$ be a minimum subset S of $V(G_i)$ satisfying all the following conditions.*

(1) $S \cap B_i = X$.

(2) $|S \cap Q| \geq k$ for every maximum clique $Q \in \mathcal{Q}(G_i) \cap \mathcal{Q}(G)$.

If $F_{k,i}(X)$ does not exist, let $F_{k,i}(X) = \Phi$ with $|\Phi| = \infty$.

Remark 3. *By Definition 9, $T = T_r$, $G = G_r$, and $\tau_{\times k}(G) = \min\{|F_{k,r}(X)| \mid X \subseteq B_r\}$.*

Lemma 4. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is a leaf node of T. Then, all sets $F_{k,i}(X)$ of G_i can be computed in $O(2^{tw(G)} \cdot tw(G))$ time.*

Proof. Since node i is a leaf node, $G_i = G[B_i]$ and $\mathcal{Q}(G_i) \cap \mathcal{Q}(G) = \mathcal{Q}(G[B_i])$. By the definition of tree decomposition, $|B_i| \leq tw(G) + 1$ and $\omega(G) = tw(G) + 1$. If B_i is not a maximum clique of G, then $F_{k,i}(X) = \Phi$ for $X \subseteq B_i$. It takes $O(tw(G)^2)$ time to check if B_i is a maximum clique. We therefore assume that B_i is a maximum clique of G. Note that $F_{k,i}(X) = X \subseteq B_i$ in this case. There are $O(2^{tw(G)})$ subsets of B_i. For each $F_{k,i}(X)$ of B_i, we verify if $F_{k,i}(X)$ has at least k vertices in B_i. The verification process can be done in $O(tw(G))$ time. Following the discussion above, all sets $F_{k,i}(X)$ of G_i can be computed in $O(2^{tw(G)} \cdot tw(G))$ time. □

Lemma 5. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is a forget node of T. Let j be the child node of i and let $x \in B_j$ be the vertex such that $B_i = B_j \setminus \{x\}$. Let $X \subseteq B_j$ such that $s_1 = |F_{k,j}(X)|$ and $s_2 = |F_{k,j}(X \cup \{x\})|$. Then,*

$$F_{k,i}(X) = \begin{cases} F_{k,j}(X) & \text{if } s_1 \leq s_2, \\ F_{k,j}(X \cup \{x\}) & \text{if } s_2 < s_1. \end{cases}$$

Proof. Since $B_i = B_j \setminus \{x\}$, $G_i = G_j$. Then, $X = F_{k,i}(X) \cap B_i = F_{k,i}(X) \cap (B_j \setminus \{x\})$. Therefore, either $F_{k,i}(X) \cap B_j = X$ or $F_{k,i}(X) \cap B_j = X \cup \{x\}$. We have $|F_{k,i}(X)| = \min\{|F_{k,j}(X)|, |F_{k,j}(X \cup \{x\})|\}$. Hence, the lemma holds. □

Lemma 6. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is an introduce node of T. Let j be the child node of i and let x be the vertex such that $B_i = B_j \cup \{x\}$. For $X \subseteq B_i$,*

$$F_{k,i}(X) = \begin{cases} F_{k,j}(X) & \text{if } B_i \notin Q(G) \text{ and } x \notin X, \\ F_{k,j}(X \setminus \{x\}) \cup \{x\} & \text{if } B_i \notin Q(G) \text{ and } x \in X, \\ F_{k,j}(X) & \text{if } B_i \in Q(G), x \notin X, \text{ and } |X| \geq k, \\ F_{k,j}(X \setminus \{x\}) \cup \{x\} & \text{if } B_i \in Q(G), x \in X, \text{ and } |X| \geq k, \\ \Phi & \text{otherwise.} \end{cases}$$

Proof. Let d be a proper descendant of the subtree T_j. Since $x \notin B_j$, $x \notin V(G_j)$, by the third condition of Definition 5. Then, $x \notin B_d$. We consider the following two cases.

Case 1: $B_i \notin Q(G)$. Assume that there exists a set $S \subseteq B_d$ such that $S \cup \{x\}$ is a maximum clique of G. By Lemmma 3, there exists a bag B_p such that $S \cup \{x\} = B_p$. Since B_p contains the vertex x, the corresponding node p is not in T_j. Then, every tree path from node d to node p contains the nodes i and j. By Lemma 1, $S = B_j = B_i \setminus \{x\}$. Then, $B_i = S \cup \{x\}$ is a maximum clique of G, but it contradicts that $B_i \notin Q(G)$. Therefore, there is no set $S \subseteq B_d$ such that $S \cup \{x\} \in Q(G)$. We have $Q(G_i) \cap Q(G) = (Q(G_j) \cap Q(G))$. Then, $F_{k,i}(X) = F_{k,j}(X)$ if $x \notin X$. Otherwise, $F_{k,i}(X) = F_{k,j}(X \setminus \{x\}) \cup \{x\}$.

Case 2: $B_i \in Q(G)$. Assume that there exists a set $S \subseteq B_d$ such that $S \cup \{x\} \in Q(G)$. By the arguments similar to those for proving Case 1, $B_i = S \cup \{x\}$. Therefore, $Q(G_i) \cap Q(G) = (Q(G_j) \cap Q(G)) \cup \{B_i\}$. Conversely, we assume that there is no set $S \subseteq B_d$ such that $S \cup \{x\} \in Q(G)$. Clearly, $Q(G_i) \cap Q(G) = (Q(G_j) \cap Q(G)) \cup \{B_i\}$. Since $X \subseteq B_i$, $|X| \geq k$. If $|X| < k$, then $F_{k,i}(X)$ does not exist. Therefore, $F_{k,i}(X) = F_{k,j}(X)$ if $x \notin X$ and $|X| \geq k$. If $x \in X$ and $|X| \geq k$, then $F_{k,i}(X) = F_{k,j}(X \setminus \{x\}) \cup \{x\}$. □

Remark 4. *The method used for proving Lemma 6 is similar to the one adopted in estimating Roman domination [17].*

Lemma 7. *Let G be a graph of bounded treewidth with a nice tree decomposition $(T, \{B_i \mid i \in T\})$. Suppose that node i is a join node of T. Let j and ℓ be the child nodes of i. For $X \subseteq B_i$,*

$$F_{k,i}(X) = F_{k,j}(X) \cup F_{k,\ell}(X).$$

Proof. Since node i is a join node, $B_i = B_j = B_\ell$. Then, $Q(G_i) = Q(G_j) \cup Q(G_\ell)$. Let $S = F_{k,j}(X) \cup F_{k,\ell}(X)$. Clearly, $S \cap B_i = X$ and $|S \cap Q| \geq k$ for $Q \in Q(G_i) \cap Q(G)$. We have $|F_{k,i}(X)| \leq |S| = |F_{k,j}(X) \cup F_{k,\ell}(X)|$.

Let $S_1 = F_{k,i}(X) \cap V(G_j)$ and $S_2 = F_{k,i} \cap V(G_\ell)$. Since X is a subset of $S_1 \cup S_2$, $|F_{k,i}(X)| = |(S_1 \cup X) \cup (S_2 \cup X)|$. Let $S_j = S_1 \cup X$ and $S_\ell = S_2 \cup X$. Clearly, $S_j \cap B_j = X$ and $S_\ell \cap B_\ell = X$. Futhermore, $|S_j \cap Q| \geq k$ for $Q \in Q(G_j) \cap Q(G)$ and $|S_\ell \cap Q| \geq k$ for $Q \in Q(G_\ell) \cap Q(G)$. Therefore, $|F_{k,j}(X)| \leq |S_j|$ and $|F_{k,\ell}(X)| \leq |S_\ell|$. We have $|F_{k,j}(X) \cup F_{k,\ell}(X)| \leq |S_j \cup S_\ell| = |F_{k,i}(X)|$. Following the discussion above, $|F_{k,i}(X)| = |F_{k,j}(X) \cup F_{k,\ell}(X)|$ and the lemma holds. □

Theorem 6. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$ and $O(|V(G)|)$ nodes. The k-fold maximum-clique transversal problem can be solved in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time.*

Proof. Assume that T is rooted at r. Then, $T = T_r$, $G = G_r$, and $\tau_{\times k}(G) = \min\{|F_{k,r}(X)| \mid X \subseteq B_r\}$. Our algorithm works from the leaves in T up to the root, computing the solutions $F_i(X)$ for each visited node i on the way through the dynamic programming technique. For all $X \in B_i$, the solutions can be computed in $O(2^{tw(G)} \cdot tw(G))$ time by Lemmas 4–7. Since T contains $O(|V(G)|)$ nodes, the k-fold maximum-clique transversal problem can be solved in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time. □

4.2. The $\{k\}$-Maximum-Clique Transversal Problem

This section studies the $\{k\}$-maximum-clique transversal problem on graphs of bounded treewidth and solves the problem with reduction and the dynamic programming technique.

4.2.1. Problem Solving by Reduction

The section shows a reduction from the $\{k\}$-maximum-clique transversal problem to the k-fold maximum-clique transversal problem by the concept of the *strong product* of two graphs.

Definition 10. *The* strong product $G \boxtimes H$ *is defined on the vertex set* $V(G) \times V(H)$*, where two vertices* (u_1, v_1) *and* (u_2, v_2) *are adjacent if and only if* $u_1 = u_2$ *and* $v_1v_2 \in E(H)$*, or* $v_1 = v_2$ *and* $u_1u_2 \in E(G)$*, or* $v_1v_2 \in E(H)$ *and* $u_1u_2 \in E(G)$*.*

Figure 2 shows the strong product of a graph G and a complete graph H. The graph $G \boxtimes H$, as shown in Figure 2c, consists of twelve vertices. From left to right, the vertices in row i are (u_i, v_1), (u_i, v_2), and (u_i, v_3) for $i \in \{1, 2, 3, 4\}$.

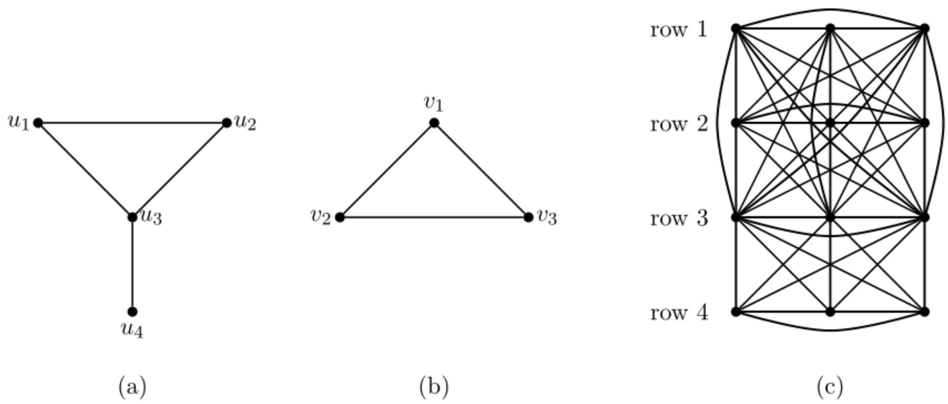

Figure 2. (**a**) A graph G with two maximal cliques. (**b**) A complete graph H. (**c**) The strong product of G and H.

Lemma 8. *Let G and H be graphs. Then,* $Q(G \boxtimes H) = \{V(G[Q_1] \boxtimes H[Q_2]) \mid Q_1 \in Q(G), Q_2 \in Q(H)\}$*.*

Proof. Let $Q_1 \in Q(G)$ and $Q_2 \in Q(H)$. The vertex set, $V(G[Q_1] \boxtimes H[Q_2])$, is a clique of $G \boxtimes H$. We have $\omega(G) \cdot \omega(H) \leq \omega(G \boxtimes H)$.

Let $V(G) = \{u_1, \ldots, u_n\}$ and $V(H) = \{v_1, \ldots, v_k\}$. Let $Q \in Q(G \boxtimes H)$. For any two distinct vertices $(u_p, v_x), (u_q, v_y) \in Q$, (1) $u_p = u_q$ and $v_xv_y \in E(H)$, or (2) $u_pu_q \in E(G)$ and $v_x = v_y$, or (3) $u_pu_q \in E(G)$ and $v_xv_y \in E(H)$. Therefore, $u_pu_q \in E(G)$ if $u_p \neq u_q$ for any two distinct vertices $(u_p, v_x), (u_q, v_y) \in Q$. Similarly, $v_xv_y \in E(H)$ if $v_x \neq v_y$ for any two distinct vertices $(u_p, v_x), (u_q, v_y) \in Q$, Then, $C_1 = \{u_i \mid (u_i, v_j) \in Q, 1 \leq i \leq n, 1 \leq j \leq k\}$ is a clique of G and $C_2 = \{v_j \mid (u_i, v_j) \in Q, 1 \leq i \leq n, 1 \leq j \leq k\}$ is a clique of H. Let $Q_1 \in Q(G)$ and $Q_2 \in Q(H)$ such that $C_1 \subseteq Q_1$ and $C_2 \subseteq Q_2$. Hence, $Q \subseteq V(G[C_1] \boxtimes H[C_2]) \subseteq V(G[Q_1] \boxtimes H[Q_2])$. We have $\omega(G \boxtimes H) \leq \omega(G) \cdot \omega(H)$. Therefore, $\omega(G \boxtimes H) = \omega(G) \cdot \omega(H)$ by the discussion above, and the lemma holds. □

Theorem 7. *Let k be a fixed positive integer and let H be a complete graph with k vertices. For any graph G and $k \geq 1$,* $\tau_{\{k\}}(G) = \tau_{\times k}(G \boxtimes H)$*.*

Proof. Let $V(G) = \{u_1, \ldots, u_n\}$ and $V(H) = \{v_1, \ldots, v_k\}$. By Lemma 8, $Q(G \boxtimes H) = \{V(G[Q] \boxtimes H) \mid Q \in Q(G)\}$. Clearly, $|Q(G \boxtimes H)| = |Q(G)|$ and $\omega(G \boxtimes H) = k \cdot \omega(G)$.

Let f a $\{k\}$-maximum-clique transversal function of G of minimum weight. Let D be a subset of $V(G \boxtimes H)$ defined by

$$D = \bigcup_{i=1}^{n} \{(u_i, v_j) \mid j = 1, 2, \ldots, f(u_i)\}.$$

Since f is a $\{k\}$-maximum-clique transversal function, $0 \leq f(u_i) \leq k$ for $1 \leq i \leq n$. Indeed, the set D exists, and $|D| = \sum_{i=1}^{n} f(u_i) = \tau_{\{k\}}(G)$. Let $C = \{u_{b_i} \mid 1 \leq i \leq \omega(G)\}$ be a maximum clique of G such that $V(G[C] \boxtimes H) = \{(u_{b_i}, v_j) \mid 1 \leq i \leq \omega(G), 1 \leq j \leq k\}$ is a maximum clique of $G \boxtimes H$. Since $f(C) = \sum_{i=1}^{\omega(G)} f(u_{b_i}) \geq k$,

$$|D \cap V(G[C] \boxtimes H)| = |\{(u_{b_i}, v_j) \mid 1 \leq i \leq \omega(G), 1 \leq j \leq f(u_{b_i})\}| = \sum_{i=1}^{\omega(G)} f(u_{b_i}) \geq k.$$

The set D is a k-fold maximum-clique transversal set of $G \boxtimes H$. We have $\tau_{\times k}(G \boxtimes H) \leq \tau_{\{k\}}(G)$.

In the following, we show that $\tau_{\{k\}}(G) \leq \tau_{\times k}(G \boxtimes H)$. Let S be a k-fold maximum-clique transversal set of $G \boxtimes H$ such that $|S| = \tau_{\times k}(G \boxtimes H)$.

Let $f : V(G) \to \{0, 1, \ldots, k\}$ be a function of G such that $f(u_i) = |S \cap \{(u_i, v_j) \mid j = 1, \ldots, k\}|$ for $1 \leq i \leq n$. Clearly, $f(V(G)) = \tau_{\times k}(G \boxtimes H)$ and $0 \leq f(u_i) \leq k$ for $1 \leq i \leq n$. Recall that $Q(G \boxtimes H) = \{V(G[C] \boxtimes H) \mid C \in Q(G)\}$. Let $C = \{u_{b_i} \mid 1 \leq i \leq \omega(G)\}$ be a maximum clique of G such that $Q = V(G[C] \boxtimes H)$ is a maximum clique of $G \boxtimes H$. Then, $Q = \{(u_{b_i}, v_j) \mid 1 \leq i \leq \omega(G), 1 \leq j \leq k\}$.

Since S is a k-fold maximum-clique transversal set of $G \boxtimes H$, $|S \cap Q| = |S \cap \{(u_{b_i}, v_j) \mid 1 \leq i \leq \omega(G), 1 \leq j \leq k\}| \geq k$. Therefore,

$$f(C) = \sum_{i=1}^{\omega(G)} f(u_{b_i}) = \sum_{i=1}^{\omega(G)} |S \cap \{(u_{b_i}, v_j) \mid 1 \leq j \leq k\}| = |S \cap Q| \geq k.$$

The function f is a $\{k\}$-maximum-clique transversal function of G. We have $\tau_{\{k\}}(G) \leq f(V(G)) = \tau_{\times k}(G \boxtimes H)$. Hence, $\tau_{\{k\}}(G) = \tau_{\times k}(G \boxtimes H)$ by the discussion above. $\square$

Lemma 9. *Let k be a fixed positive integer and let H be a complete graph with k vertices. Given a nice tree decomposition $D = (T, B = \{B_i \mid i \in T\})$ of a graph G of width $tw(G)$ and $O(|V(G)|)$ nodes, one can find a nice tree decomposition of $G \boxtimes H$ of width $k(tw(G) + 1) - 1$ and with at most $4 \cdot |V(G \boxtimes H)|$ nodes in $O(|V(G \boxtimes H)|)$ time.*

Proof. Let $V(G) = \{u_i \mid 1 \leq i \leq n\}$ and $V(H) = \{v_i \mid 1 \leq i \leq k\}$. Let $B'_i = \{(u_\ell, v_j) | u_\ell \in B_i, 1 \leq j \leq k\}$ and $D' = (T, B' = \{B'_i \mid i \in T\})$. We show as follows that D' is a tree decomposition of $G \boxtimes H$.

(1) Every vertex $u_\ell \in V(G)$ appears in at least one bag of B. Suppose that $u_\ell \in B_i$. The vertices (u_ℓ, v_1), $(u_\ell, v_2), \ldots, (u_\ell, v_k)$ all appear in $B'_i \in B'$. Therefore, every vertex $(u_\ell, v_j) \in V(G \boxtimes H)$ appears in at least one bag $B'_i \in B'$.

(2) For any two adjacent vertices $(u_x, v_p), (u_y, v_q) \in V(G \boxtimes H)$, $u_x = u_y$ and $v_p v_q \in E(H)$, or $u_x u_y \in E(G)$ and $v_p = v_q$, or $u_x u_y \in E(G)$ and $v_p v_q \in E(H)$. Consider u_x and u_y. Either $u_x = u_y$ or $u_x u_y \in E(G)$. If $u_x = u_y$, then u_x and u_y are in the same bags. Suppose that $u_x \neq u_y$ and $u_x u_y \in E(G)$. Since there is at least one bag of B containing both vertices of e for every edge $e \in E(G)$, at least one bag contains the vertices u_x and u_y. Let $u_x, u_y \in B_i$. The vertices (u_x, v_1), $(u_x, v_2), \ldots, (u_x, v_k)$ and (u_y, v_1), $(u_y, v_2), \ldots, (u_y, v_k)$ are in $B'_i \in B'$. The bag B'_i contains (u_x, v_p) and (u_y, v_q). Therefore, there is at least one bag of B' containing every pair of adjacent vertices of $G \boxtimes H$.

(3) Suppose that (u_ℓ, v_j) is a vertex of $G \boxtimes H$ and it appears in two bags $B'_i, B'_p \in B'$. Let j be a node on the tree path from node i to node p in T. Since (u_ℓ, v_j) appears in B'_i

and B'_p, we know that $u_\ell \in B_i$ and $u_\ell \in B_p$. Then, $u_\ell \in B_j$ by Lemma 1. The vertices (u_ℓ, v_1), $(u_\ell, v_2), \ldots, (u_\ell, v_k)$ are in B'_j. Clearly, $(u_\ell, v_j) \in B'_j$. Hence, a vertex (u_ℓ, v_j) appears in every bag B'_j for the node j on the tree path from node i to node p in T if it appears in $B'_i, B'_j \in B'$.

Following the discussion above, $D' = (T, B')$ is a tree decomposition of $G \boxtimes H$ of width $k(tw(G)+1)-1$ and with $O(|V(G)|)$ nodes. By Lemma 2, we can obtain a nice tree decomposition of $G \boxtimes H$ of width $k(tw(G)+1)-1$ with at most $4 \cdot |V(G \boxtimes H)|$ nodes in $O(|V(G \boxtimes H))$ time. □

Theorem 8. *Let G be a graph of bounded treewidth. Given a nice tree decomposition $D = (T, B = \{B_i \mid i \in T\})$ of G of width $tw(G)$ and $O(|V(G)|)$ nodes, the $\{k\}$-maximum-clique transversal problem can be solved in $O(2^{k(tw(G)+1)} \cdot k^2 \cdot (tw(G)+1)|V(G)|)$ time.*

Proof. Let H be a complete graph of k vertices. By Lemma 9, we can obtain a nice tree decomposition of $G \boxtimes H$ of width $k(tw(G)+1)-1$ with at most $4 \cdot |V(G \boxtimes H)$ nodes in $O(|V(G \boxtimes H)|)$ time. Since k is fixed and G is a graph of bounded treewidth, $G \boxtimes H$ is a graph of bounded treewidth $k(tw(G)+1)-1$. Note that $|V(G \boxtimes H)| = k \cdot |V(G)|$. By Theorems 6 and 7, the $\{k\}$-maximum-clique transversal problem can be solved in $O(2^{k(tw(G)+1)} \cdot k^2 \cdot (tw(G)+1)|V(G)|)$ time. □

4.2.2. Problem Solving by Dynamic Programming

This section studies the $\{k\}$-maximum-clique transversal problem by the dynamic programming technique for graphs of bounded treewidth.

Definition 11. *Assume that G is a graph and k is a positive integer. Let $X = (X_0, X_1, \ldots, X_k)$ be a $(k+1)$-tuple of subsets of $V(G)$. The weight of the $(k+1)$-tuple X, denoted by $w(X)$, is $\sum_{i=0}^{k}(|X_i| \cdot i)$. Let $\ell \in \{0, 1, \ldots, k\}$ and let $V' \subseteq V(G)$. Let $Y = (Y_0, Y_1, \ldots, Y_k)$ be another $(k+1)$-tuple of subsets of $V(G)$. We give the following notations and definitions.*

1. *$X \oplus Y$ denotes the $(k+1)$-tuple $(X_0 \cup Y_0, \ldots, X_k \cup Y_k)$.*
2. *$X \oplus V'$ denotes the $(k+1)$-tuple $(X_0 \cup V', \ldots, X_k \cup V')$.*
3. *$X \oplus_\ell V'$ denotes the $(k+1)$-tuple $(S_0, \ldots, S_k)$ such that $S_\ell = X_\ell \cup V'$ and $S_i = X_i$ for $i \in \{0, 1, \ldots, k\} \setminus \{\ell\}$.*
4. *$X \otimes V'$ denotes the $(k+1)$-tuple $(X_0 \cap V', \ldots, X_k \cap V')$.*
5. *$X \ominus_\ell V'$ denotes the $(k+1)$-tuple $(S_0, \ldots, S_k)$ such that $S_\ell = X_\ell \setminus V'$ and $S_i = X_i$ for $i \in \{0, 1, \ldots, k\} \setminus \{\ell\}$.*
6. *A $(k+1)$-tuple $S = (S_0, \ldots, S_k)$ is a $(k+1)$-partition of $V(G)$ satisfying the following conditions.*
   (a) $\bigcup_{i=0}^{k} S_i = V(G)$.
   (b) $S_i \cap S_j = \emptyset$ *for* $0 \le i < j \le k$.
7. *A $(k+1)$-assignment of $V(G)$ is a $(k+1)$-partition $F = (F_0, \ldots, F_k)$ of $V(G)$ such that $w(F \otimes Q) \ge k$ for every $Q \in Q(G)$.*

Remark 5. *For a fixed positive integer k, a $\{k\}$-maximum clique transversal function of a graph G can be regarded as a $(k+1)$-assignment of $V(G)$, and vice versa. Then, $\tau_{\{k\}}(G) = \min\{w(S) \mid S$ is a $(k+1)$-assignment of $V(G)\}$.*

Definition 12. *Let k be a fixed positive integer and let G be a graph of bounded treewidth with a nice tree decomposition $(T, \{B_i \mid i \in T\})$ rooted at node r. For each node $i \in T$, let $X = (X_0, \ldots, X_k)$ be a $(k+1)$-partition of B_i and let $F_i(X)$ be a $(k+1)$-assignment $S = (S_0, \ldots, S_k)$ of $V(G_i)$ of minimum weight satisfying all the following conditions.*

(1) $S \otimes B_i = X$.
(2) $w(S \otimes Q) \ge k$ *for every* $Q \in Q(G_i) \cap Q(G)$.

If $F_i(X)$ does not exist, let $F_i(X) = \Phi$ with $w(\Phi) = \infty$.

Remark 6. *By Definition 12, $T = T_r$, $G = G_r$, and $\tau_{\{k\}}(G) = \min\{w(F_r(X)) \mid X$ is a $(k+1)$-partition of $B_r\}$.*

Lemma 10. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is a leaf node of T. For all $(k+1)$-partitions X of B_i, $F_i(X)$ of G_i can be computed in $O((k+1)^{tw(G)+1} \cdot tw(G))$ time.*

Proof. Since node i is a leaf node, $G_i = G[B_i]$ and $Q(G_i) \cap Q(G) = Q(G[B_i])$. By the definition of tree decomposition, $|B_i| \le tw(G) + 1$ and $\omega(G) = tw(G) + 1$. If B_i is not a maximum clique of G, then $F_i(X) = \Phi$ for every $(k+1)$-partition X of B_i. It takes $O(tw(G)^2)$ time to check if B_i is a maximum clique. We therefore assume that B_i is a maximum clique of G. The number of $(k+1)$-assignments of B_i is $(k+1)^{tw(G)+1}$. For each $(k+1)$-assignment $F_i(X)$ of B_i, we verify if $w(F_i(X)) \ge k$. The verification process can be done in $O(tw(G))$ time. Following the discussion above, all $(k+1)$-assignments $F_i(X)$ of G_i can be computed in $O((k+1)^{tw(G)+1} \cdot tw(G))$ time. □

Lemma 11. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is a forget node of T. Let j be the child node of i and let $x \in B_j$ such that $B_i = B_j \setminus \{x\}$. Let X be a $(k+1)$-partition of B_i and $Y = \{(X \oplus_\ell \{x\}) \mid 0 \le \ell \le k\}$. Let $Z \in Y$ such that $w(F_j(Z)) = \min\{w(F_j(X')) \mid X' \in Y\}$. Then, $F_i(X) = F_j(Z)$.*

Proof. Since $B_i = B_j \setminus \{x\}$, $G_i = G_j$. The $(k+1)$-assignment $F_i(X)$ is also a $(k+1)$-assignment of $V(G_j)$. Let $X' = F_i(X) \otimes B_j$. Then, X' is a $(k+1)$-partition of B_j. There exists exactly one integer $\ell \in \{0, 1, \ldots, k\}$ such that $X' = (X \otimes_\ell \{x\})$. Let $Y = \{(X \oplus_\ell \{x\}) \mid 0 \le \ell \le k\}$. Hence, $F_i(X) = F_j(Z)$, where $Z \in Y$ and $w(F_j(Z)) = \min\{w(F_j(X')) \mid X' \in Y\}$. □

Lemma 12. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$. Suppose that node i is an introduce node of T. Let j be the child node of i and let x be the vertex such that $B_i = B_j \cup \{x\}$. Let $\ell \in \{0, 1, \ldots, k\}$ and let $X = (X_0, \ldots X_k)$ be a $(k+1)$-partition of B_i such that $x \in X_\ell$. Then,*

$$F_i(X) = \begin{cases} F_j(X \ominus_\ell \{x\}) \oplus_\ell \{x\} & \text{if either } B_i \notin Q(G) \text{, or } B_i \in Q(G) \text{ and } w(X) \ge k, \\ \Phi & \text{otherwise.} \end{cases}$$

Proof. Let d be a proper descendant of the subtree T_j. Since $x \notin B_j$, $x \notin V(G_j)$, by the third condition of Definition 5. Then, $x \notin B_d$. We consider the following two cases.

Case 1: $B_i \notin Q(G)$. By the arguments for the discussion of Case 1 in the proof of Lemma 6, $Q(G_i) \cap Q(G) = Q(G_j) \cap Q(G)$. For any $(k+1)$-partition X' of B_j, $w(F_j(X') \otimes Q)) \ge k$ for every $Q \in Q(G_i) \cap Q(G)$.

Let $\ell \in \{0, 1, \ldots, k\}$ and let X be a $(k+1)$-partition of B_i such that $X = X' \oplus_\ell \{x\}$. Then, $F_j(X') \oplus_\ell \{x\}$ is a $(k+1)$-assignment S of $V(G_i)$ such that $S \otimes B_i = X$. Therefore, $w(F_i(X)) \le w(F_j(X') \oplus_\ell \{x\})$.

Conversely, let $X = (X_0, X_1, \ldots, X_k)$ be a $(k+1)$-partition of B_i such that $x \in X_\ell$. Then, $F_i(X) \ominus_\ell \{x\}$ is a $(k+1)$-assignment S' of $V(G_j)$ such that $S' \otimes B_j = X \ominus_\ell \{x\}$. Let $X' = X \ominus_\ell \{x\}$. Therefore, $w(F_i(X)) \ge w(F_j(X') \oplus_\ell \{x\})$.

Following the discussion above, we know that $F_i(X) = F_j(X \ominus_\ell \{x\}) \oplus_\ell \{x\}$.

Case 2: $B_i \in Q(G)$. By the arguments for the discussion of Case 2 in the proof of Lemma 6, $Q(G_i) \cap Q(G) = (Q(G_j) \cap Q(G)) \cup \{B_i\}$. Since $B_i \in Q(G)$ and $F_i(X) \otimes B_i = X$, we have $w(X) \ge k$. If $w(X) < k$, then $F_i(X)$ does not exist. Following the arguments similar to those for proving Case 1, we know that $w(F_i(X)) = w(F_j(X') \oplus_\ell \{x\})$. Therefore, $F_i(X) = F_j(X \ominus_\ell \{x\}) \oplus_\ell \{x\}$. □

Lemma 13. *Let G be a graph of bounded treewidth with a nice tree decomposition $(T, \{B_i \mid i \in T\})$. Suppose that node i is a join node of T. Let j and ℓ be the child nodes of i. For each $(k+1)$-partition $X = (X_0, X_1, \ldots, X_k)$ of B_i,*

$$F_i(X) = F_j(X) \oplus F_\ell(X).$$

Proof. Since node i is a join node, $B_i = B_j = B_\ell$. Then, $Q(G_i) = Q(G_j) \cup Q(G_\ell)$. Let $S = F_j(X) \oplus F_\ell(X)$. Clearly, $S \otimes B_i = X$ and $w(S \otimes Q) \geq k$ for $Q \in Q(G_i) \cap Q(G)$. We have $w(F_i(X)) \leq w(S) = w(F_j(X) \oplus F_\ell(X))$.

Let $S_1 = F_i(X) \otimes V(G_j)$ and $S_2 = F_i(X) \otimes V(G_\ell)$. Then, S_1 and S_2 are $(k+1)$-assignments of $V(G_j)$ and $V(G_j)$, respectively. Furthermore, $S_1 \otimes B_j = S_2 \otimes B_\ell = X$. Therefore, $w(F_j(X) \oplus F_\ell(X)) \leq w(S_1 \oplus S_2) = w(F_i(X))$.

Following the discussion above, $w(F_i(X)) = w(F_j(X) \oplus F_\ell(X))$ and the lemma holds. □

Theorem 9. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$ and $O(|V(G)|)$ nodes. The $\{k\}$-maximum-clique transversal problem can be solved in $O((k+1)^{tw(G)+1} \cdot tw(G) \cdot |V(G)|)$ time.*

Proof. Assume that T is rooted at r. Then, $T = T_r$, $G = G_r$, and $\tau_{\times k}(G) = \min\{w(F_r(X)) \mid X$ is a $(k+1)$-partition of $B_r\}$. Our algorithm works from the leaves in T up to the root, computing the solutions $F_i(X)$ for each visited node i on the way through the dynamic programming technique. For all $(k+1)$-partitions X of B_i, the solutions can be computed in $O((k+1)^{tw(G)+1} \cdot tw(G))$ time by Lemmas 10–13. Since T contains $O(|V(G)|)$ nodes, the $\{k\}$-maximum-clique transversal problem can be solved in $O((k+1)^{tw(G)+1} \cdot tw(G) \cdot |V(G)|)$ time. □

4.3. The Signed and Minus Maximum-Clique Transversal Problems

The section deals with the signed and minus maximum-clique transversal problems on graphs of bounded treewidth.

Theorem 10. *Assume that $G = (V, E)$ is a graph and $\omega(G)$ is fixed. Let $k = \lfloor \omega(G)/2 \rfloor + 1$. Then, $\tau^s_M(G) = 2\tau_{\times k}(G) - n$.*

Proof. The theorem holds by Theorem 7 in Lee's article [3]. □

Theorem 11. *Let $D = (T, \{B_i \mid i \in T\})$ be a nice tree decomposition of a graph G of width $tw(G)$ and $O(|V(G)|)$ nodes. The signed maximum-clique transversal problem can be solved in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time.*

Proof. It follows from Theorems 6 and 10. □

We introduce the 2-clique graph extension and positive maximum-clique transversal functions as follows.

Definition 13. *The 2-clique graph extension H of a graph G is defined as follows.*

1. $V(H) = \{v_1, v_2 \mid v \in V(G)\}$.
2. $E(H) = \{v_1 v_2 \mid v \in V(G)\} \cup \{u_1 v_1, u_1 v_2, u_2 v_1, u_2 v_2 \mid uv \in E(G)\}$.

Figure 3 shows an example of the 2-clique graph extension H of a graph G.

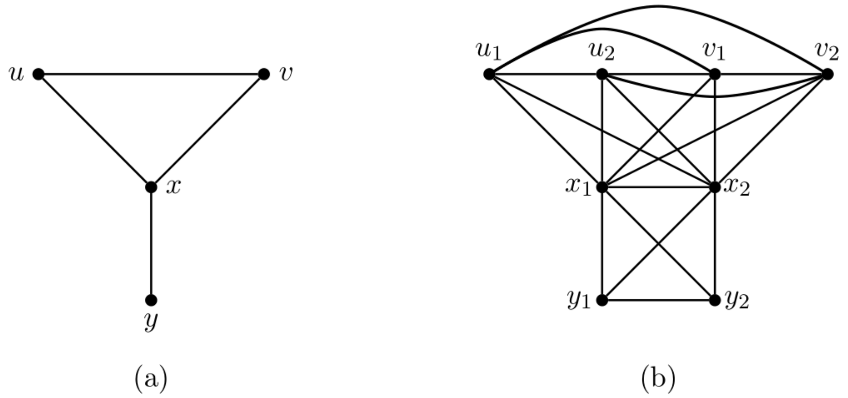

Figure 3. (**a**) A graph G with two maximal cliques. (**b**) The 2-clique graph extension H of G.

Definition 14. *Assume that G is a graph. A function* $f : V(G) \to \{0, 1, 2\}$ *is a positive maximum-clique transversal function of G if* $f(Q) \geq \omega(G) + 1$ *for* $Q \in \mathcal{Q}(G)$. *The positive maximum-clique transversal number of G, denoted by* $\tau_M^+(G)$, *is the minimum weight of a positive maximum-clique transversal function of G. The positive maximum-clique transversal problem is to find a positive maximum-clique transversal function of G of minimum weight.*

Lemma 14. *For any graph G,* $\tau_M^+(G) = \tau_M^-(G) + |V(G)|$.

Proof. Let f be a minus maximum-clique transversal function of G of minimum weight. We define a function h of G by $h(v) = f(v) + 1$ for every $v \in V(G)$. Since $f(v) \in \{-1, 0, 1\}$, $h(v) \in \{0, 1, 2\}$ for $v \in V(G)$. Let Q be a maximum clique of G. Then $h(Q) = \sum_{v \in Q}(f(v) + 1) = f(Q) + |Q| \geq 1 + \omega(G)$. The function h is a positive maximum-clique function of G. We have $\tau_M^+(G) \leq \tau_M^-(G) + |V(G)|$.

Conversely, let h be a positive maximum-clique transversal function of G of minimum weight. We define a function f of G by $f(v) = h(v) - 1$ for every $v \in V(G)$. Since $h(v) \in \{0, 1, 2\}$, $f(v) \in \{-1, 0, 1\}$ for every $v \in V(G)$. Let Q be a maximum clique of G. Then $f(Q) = \sum_{v \in Q}(h(v) - 1) = h(Q) - |Q| \geq 1$. The function f is a minus maximum-clique function of G. We have $\tau_M^-(G) \leq \tau_M^+(G) - |V(G)|$. Following what we have discussed above, we know that $\tau_M^+(G) = \tau_M^-(G) + |V(G)|$. $\square$

Lemma 15. *Assume that G is a graph with fixed clique number* $\omega(G)$. *Let H be the 2-clique graph extension of G and* $k = \omega(G) + 1$. *Then,* $\tau_M^+(G) = \tau_{\times k}(H)$.

Proof. Let $\mathcal{Q}(G) = \{C_1, \ldots, C_\ell\}$ and let $Q_i = \{v_1, v_2 \mid v \in C_i\}$ for $1 \leq i \leq \ell$. By the construction of H, it can be easily verified that $\mathcal{Q}(H) = \{Q_1, \ldots, Q_\ell\}$.

Let f be a positive maximum-clique transversal function of G of minimum weight. We define a subset D of $V(H)$ as follows.

1. For any vertex $v \in V(G)$ with $f(v) = 2$, D includes both of the vertices v_1 and v_2.
2. For any vertex $v \in V(G)$ with $f(v) = 1$, D contains precisely one of the vertices v_1 and v_2.
3. For any vertex $v \in V(G)$ with $f(v) = 0$, D comprises none of the vertices v_1 and v_2.

Let $i \in \{1, \ldots, \ell\}$. Then,

$$|D \cap Q_i| = \sum_{v \in C_i} |D \cap \{v_1, v_2\}| = \sum_{v \in C_i} f(v) = f(C_i) \geq \omega(G) + 1 \geq k.$$

The set D is a k-fold maximum-clique transversal set of H. Hence, $\tau_{\times k}(H) \leq \tau_M^+(G)$.

Conversely, let S be a k-fold maximum-clique transversal set of H with $\tau_{\times k}(H)$ vertices. We define a function h of G as follows.

1. For any two vertices $v_1, v_2 \in V(H)$, $h(v) = 2$ if S includes both of them.

2. For any two vertices $v_1, v_2 \in V(H)$, $h(v) = 1$ if S contains precisely one of them.
3. For any two vertices $v_1, v_2 \in V(H)$, $h(v) = 0$ if S comprises none of them.

   Let $i \in \{1, \ldots, \ell\}$. Then,

$$h(C_i) = \sum_{v \in C_i} h(v) = \sum_{v \in C_i} |S \cap \{v_1, v_2\}| = |S \cap Q_i| \geq k \geq \omega(G) + 1.$$

The function h is a positive maximum-clique transversal function of G. Hence, $\tau_M^+(G) \leq \tau_{\times k}(H)$. Following the discussion above, we have $\tau_M^+(G) = \tau_{\times k}(H)$. □

Theorem 12. *Assume that G is a graph with fixed clique number $\omega(G)$. Let H be the 2-clique graph extension of G and $k = \omega(G) + 1$. Then, $\tau_M^-(G) = \tau_{\times k}(H) - |V(G)|$.*

Proof. The theorem follows from Lemmas 14 and 15. □

Lemma 16. *Let G be a graph of bounded treewidth nd let H be a 2-clique graph extension of G. Given a nice tree decomposition $D = (T, B = \{B_i \mid i \in T\})$ of G of width $tw(G)$ and $O(|V(G)|)$ nodes, one can find a nice tree decomposition of H of width $2(tw(G) + 1)$ and with at most $4|V(H)|$ nodes in $O(|V(H)|)$ time.*

Proof. Let $B_i' = \{v_1, v_2 | v \in B_i\}$ for $B_i \in B$ and $D' = (T, B' = \{B_i' \mid i \in T\})$. We show as follows that D' is a tree decomposition of H.

(1) Every vertex $v \in V(G)$ appears in at least one bag of B. Suppose that $v \in B_i$. Since $V(H) = \{v_1, v_2 \mid v \in V(G)\}$, $v_1, v_2 \in B_i'$. Therefore, every vertex of H appears in at least one bag $B_i' \in B'$.

(2) For each edge $e = uv \in E(G)$, there is at least one bag of B containing the vertices u and v. Suppose that $u, v \in B_i$. Since $E(H) = \{v_1v_2 \mid v \in V(G)\} \cup \{u_1v_1, u_1v_2, u_2v_1, u_2v_2 \mid uv \in E(G)\}$, u_1, u_2, v_1, and v_2, are in the bag B_i'. Therefore, there is at least one bag of B' containing both x and y for all edges $xy \in E(H)$.

(3) Let $v \in V(G)$. Then, $v_1, v_2 \in V(H)$. Clearly, $v_1, v_2 \in B_i'$ if and only if $v \in B_i$. By Lemma 1, if v appears in two bag $B_p, B_q \in B$, then it appears in every bag B_j for the node j on the tree path from node p to node q in T. Since $V(H) = \{v_1, v_2 \mid v \in V(G)\}$, v_1 and v_2 are both in B_p' and B_q' and they appear in every bag B_j' for the node j on the tree path from node p to node q in T.

Following the discussion above, $D' = (T, B')$ is a tree decomposition of H of width $2(tw(G) + 1)$ and with $O(|V(G)|)$ nodes. By Lemma 2, we can obtain a nice tree decomposition of H of width $2(tw(G) + 1)$ with at most $4|V(H)|$ nodes in $O(|V(H))$ time. □

Theorem 13. *Let G be a graph of bounded treewidth. Given a nice tree decomposition $D = (T, B = \{B_i \mid i \in T\})$ of G of width $tw(G)$ and $O(|V(G)|)$ nodes, the minus maximum-clique transversal problem can be solved in $O(4^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time.*

Proof. Let H be the 2-clique graph extension of G. By Lemma 16, we can obtain a nice tree decomposition of H of width $2(tw(G) + 1) - 1$ with at most $4|V(H)|$ nodes in $O(|V(H)|)$ time. Since G is a graph of bounded treewidth, H is a graph of bounded treewidth $2(tw(G) + 1) - 1$. Note that $|V(H)| = 2 \cdot |V(G)|$. By Theorems 6 and 7, the minus maximum-clique transversal function problem can be solved in $O(4^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time. □

Remark 7. *The complexity of the minus maximum-clique transversal problem for graphs of bounded treewidth can be improved to $O(3^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ by our dynamic programming technique used for the $\{k\}$-maximum-clique transversal problem. We leave it to interested readers.*

5. Conclusions

In this paper, we have shown that the k-fold maximum-clique transversal problem can be solved in $O(2^{tw(G)} \cdot tw(G) \cdot |V(G)|)$ time for any graph G with bounded treewidth $tw(G)$. The maximum-clique transversal problem is a particular case of the k-fold maximum-clique transversal problem and the $\{k\}$-maximum-clique transversal problem with $k = 1$. Therefore, the problem can be solved in the same running time. Lokshtanov et al. [18] obtained a number of lower bounds on the running time of algorithm solving problems on graphs of bounded treewidth under the strong exponential time hypothesis (SETH) of Impagliazzo and Paturi [19]. Based on SETH, we conjecture that the maximum-clique transversal problem cannot be solved in $(2-\epsilon)^{tw(G)}|V(G)|^{O(1)}$ time for any $\epsilon > 0$. Although SETH is still somewhat controversial and not entirely accepted by the computational complexity community, the lower bound still delivers valuable messages. If the lower bound can be obtained under SETH, then the $\{k\}$-maximum-clique transversal problem cannot be solved in in $(k+1-\epsilon)^{tw(G)}|V(G)|^{O(1)}$ time for any $\epsilon > 0$. Our algorithms on graphs of bounded treewidth are probably optimal. Finally, we suggest the following open questions for future work:

1. Tables 1 and 2 show that the respective parameterized complexities of the clique transversal, k-fold maximum-clique transversal, and $\{k\}$-maximum-clique transversal problems remain unknown for planar graphs. Can we find fixed-parameter tractable algorithms to solve them for planar graphs if the considered problem is parameterized by the solution size or weight? Or is it possible to find improved algorithms for planar graphs of bounded treewidth?
2. In Table 2, all the considered problems parameterized by treewidth are fixed-parameter tractable. Can we prove that all the considered problems remain fixed-parameter tractable if given other parameters?
3. This paper considers only maximum cliques for simple graphs. In reality, dicliques also appear in more general directed graphs [20]. It is interesting to consider the problems for directed ones.

Funding: This research is supported by a Taiwanese grant under Grant No. NSC-97-2218-E-130-002-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Chang, M.S.; Kloks, T.; Lee, C.M. Maximum clique transversals. In Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science, Boltenhagen, Germany, 14–16 June 2001; Lecture Notes in Computer Science: Berlin, Germany, 2001; pp. 32–43.
2. Lee, C.-M.; Chang, M.-S. Signed and minus clique-transversal function on graphs. *Inform. Process. Lett.* **2009**, *109*, 414–417. [CrossRef]
3. Lee, C.-M. Variations of maximum-clique transversal sets on graphs. *Ann. Oper. Res.* **2010**, *181*, 21–66. [CrossRef]
4. Lee, C.-M. Weighted maximum-clique transversal sets of graphs. *ISRN Discrete Math.* **2011**, *2011*, 32–43. [CrossRef]
5. Lee, C.-M. Algorithmic Aspects of Some Variations of Clique Transversal and Clique Independent Sets on Graphs. *Algorithms* **2021**, *14*, 22. [CrossRef]
6. Liu, K.; Lu, M. Complete-Subgraph-Transversal-Sets problem on bounded treewidth graphs. *J. Combin. Optim.* **2021**, *41*, 923–933. [CrossRef]
7. Wang, H.; Kang, L.; Shan, E. Signed clique-transversal functions in graphs. *Int. J. Comput. Math.***2010**, *87*, 2398–2407. [CrossRef]
8. Wang, H.; Shan, E. The signed maximum-clique transversal number of regular graphs. *Int. J. Comput. Math.* **2012**, *89*, 741–751. [CrossRef]
9. Xu, G.; Shan, E.; Kang, L.; Chang, T.C.E. The algorithmic complexity of the minus clique-transversal problem. *Appl. Math. Comput.* **2007**, *189*, 1410–1418. [CrossRef]

10. Downey, R.G.; Fellows, M.R. *Parameterized Complexity*; Monographs in Computer Science; Springer: New York, NY, USA, 1999.
11. Alber, J.; Bodlaender, H.; Fernau, H.; Kloks, T.; Niedermeier, R. Fixed Parameter Algorithms for DOMINATING SET and Related Problems on Planar Graphs. *Algorithmica* **2002**, *33*, 461–493. [CrossRef]
12. Brandstädt, A.; Le, V.B.; Spinrad, J.P. *Graph Classes—A Survey*; SIAM Monographs on Discrete Mathematics and Applications; SIAM: Philadelphia, PA, USA, 1999.
13. Rose, D.J. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.* **1970**, *32*, 597–609. [CrossRef]
14. Kloks, T. *Treewidth—Computations and Approximations*; Lecture Notes in Computer Science: Berlin, Germany, 1993; p. 842.
15. Arnbor, S.; Proskurowski, A. Characterization and recognition of partial 3-trees. *SIAM J. Alg. Discete Meth.* **1986**, *7*, 305–314. [CrossRef]
16. Bodlaender, H.; Möhring, R.H. The pathwidth and treewidth of cographs. *SIAM J. Discrete Math.* **1993**, *6*, 181–188. [CrossRef]
17. Alhevaz, A.; Darkooti, M.; Rahbani, H.; Shang, Y. Strong Equality of Perfect Roman and Weak Roman Domination in Trees. *Mathematics* **2019**, *7*, 997. [CrossRef]
18. Lokshtanov, D.; Marx, D.; Saurabh, S. Known Algorithm on Graphs of Bounded Treewidth Are Probably Optimal. *ACM Trans. Algorithms* **2018**, *14*, 30. [CrossRef]
19. Impagliazzo, R.; Paturi, R. On the complexity of *k*-SAT. *J. Comput. Syst. Sci.* **2001**, *62*, 275–291. [CrossRef]
20. Shang, Y. Large dicliques in a directed inhomogeneous random graph. *Int. J. Comput. Math.* **2013**, *90*, 445–456. [CrossRef]