Target Set Selection Parameterized by Vertex Cover and More

Suman Banerjee · Rogers Mathew · Fahad Panolan

Accepted: 10 August 2022 / Published online: 6 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Diffusion is a natural phenomenon in many real-world networks. Spreading of ideas, rumors in an online social network; propagation of virus, malware in a computer network; spreading of diseases in a human contact network, etc. are some real-world examples of this. Diffusion often starts from a set of initial nodes known as seed nodes. A node can be in any one of the following two states: influenced (active) or not influenced (inactive). We assume that a node can change its state from inactive to active, however, not vice versa. Only the seed nodes are active initially and the information is dissipated from these seed nodes in discrete time steps. Each node \( v \) is associated with a threshold value \( \tau(v) \) which is a positive integer. A node \( v \) will be influenced at time step \( t' \), if there are at least \( \tau(v) \) number of nodes in its neighborhood which have been activated on or before time step \( t' - 1 \). The diffusion stops when no more node-activation is possible. Given a simple, undirected graph \( G \) with a threshold function \( \tau: V(G) \rightarrow \mathbb{N} \), the Target Set Selection (TSS) problem is about choosing a minimum cardinality set, say \( S \subseteq V(G) \), such that starting a diffusion process with \( S \) as its seed set will eventually result in activating all the nodes in \( G \). For any non-negative integer \( i \), we say a set \( T \subseteq V(G) \) is a degree-\( i \) modulator of \( G \) if the degree of any vertex in the graph \( G - T \) is at most \( i \). Degree-0 modulators of a graph are precisely its vertex covers. Consider a graph \( G \) on \( n \) vertices and \( m \) edges. We have the following results on the TSS problem:

- It was shown by Nichterlein et al. (Soc. Netw. Anal. Min. 3(2), 233–256 2013) that it is possible to compute an optimal-sized target set in \( O(2^{2^t+1}m) \) time, where \( t \) denotes the cardinality of a minimum degree-0 modulator of \( G \). We improve this result by designing an algorithm running in time \( 2^{O(t \log t)}n \).
- We design a \( 2^{2^{O(t)}}n^{O(1)} \) time algorithm to compute an optimal target set for \( G \), where \( t \) is the size of a minimum degree-1 modulator of \( G \).
• We show that for a graph on $n$ vertices of treewidth $s$, the TSS problem cannot be solved in $f(s)n^{o(s/\log s)}$ time unless the Exponential Time Hypothesis fails. This is an improvement over the previously known lower bound of $f(s)n^{o(\sqrt{s})}$ due to Ben-Zwi et al. (Discret. Optim. 8(1), 87–96 2011). In fact, we prove that same lower bound holds when parameterized by tree-depth or star-deletion number.

Keywords Target set selection · FPT · Vertex cover · Integer programming

1 Introduction

Diffusion is a natural phenomenon in many real-world networks such as diffusion of information, innovation, ideas, rumors in an online social network [1]; propagation of virus, malware in a computer network [2]; spreading of contaminating diseases in a human contact network [3], and many more. Depending on the situation, we want to maximize/minimize the spread. For example, in the case of propagation of information in a social network, sometimes we want to maximize the spread so that a large number of people are aware of the piece of information. On the other hand, in the case of spreading of contaminating diseases, we would want to minimize the spread. In this paper, the practical essence of our study is in and around the first situation.

Diffusion starts from a set of initial nodes known as seed nodes. A node can be in any one of the following two states: influenced (also known as active) or not influenced (also known as inactive). We assume that a node can change its state from inactive to active, however, not vice versa. Only the seed nodes are active initially and the information is disseminated in discrete time steps from these seed nodes. Each node $v$ is associated with a threshold value $\tau(v)$ which is a positive integer. A node $v$ will be influenced at time step $t'$, if it has at least $\tau(v)$ number of nodes in its neighborhood which have been activated on or before time step $(t' - 1)$. The diffusion process stops when no more node-activation is possible. A set of seed nodes is called a target set if diffusion starting from these seed nodes spreads to the entire network thereby influencing every node.

Problem Definition In our study, we assume that the social network is represented by an undirected graph $G$, where $V(G)$ and $E(G)$ are the set of vertices and edges of $G$, respectively, and there is a threshold function $\tau : V(G) \to \mathbb{N}$ that assigns a threshold value to each node. Let $S \subseteq V(G)$ be a set of seed nodes from where diffusion starts. As described in the above paragraph, influence propagates in discrete time steps, i.e., $A[S, 0] \subseteq A[S, 1] \subseteq A[S, 2] \subseteq \cdots \subseteq A[S, i] \subseteq \cdots \subseteq V(G)$, where $A[S, i]$ denotes the set of nodes that has been influenced on or before the $i^{th}$ time stamp and $A[S, 0] = S$. For all $i > 0$, the diffusion process can be expressed by the following equation:

$$A[S, i] = A[S, i - 1] \cup \{u \mid N(u) \cap A[S, i - 1] \geq \tau(u)\},$$

where $N(u)$ denotes the set of neighbors of $u$. For any seed set $S$, we define $\text{influence}_G(S) := \bigcup_{i \geq 0} A[S, i]$. Observe that the number of steps in the diffusion process is at most $|V(G)|$ as at least
one new node is activated in every time step; else the diffusion process stops. The Target Set Selection (TSS) problem is about finding a minimum cardinality target set. In other words, it is about finding a minimum cardinality seed set $S$ such that $\text{influence}_{G}(S) = V(G)$.

**Related Work** Chen [4] showed that TSS cannot be approximated within a factor of $O(2^{\log^{1-\varepsilon} n})$ of the optimum for a fixed constant $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$, by a reduction from the MINREP Problem. They also showed that TSS is NP-hard for bounded-degree bipartite graphs with a threshold value not greater than 2 at each vertex by a reduction from a variant of the 3-SAT problem. For trees, they proposed a polynomial-time exact algorithm. Chiang et al. [5] showed that TSS can be solved in linear time for block-cactus graphs with an arbitrary threshold and for chordal graph with threshold at most 2. Cicalese et al. [6] proposed exact algorithms to find minimum cardinality target set for bounded clique width and trees. Chopin et al. [7] showed that upper bounding the threshold to a constant leads to efficiently solvable instances of the TSS problem under the parameterized complexity theoretic framework. They showed that TSS is $W[1]$-hard with respect to the parameters feedback vertex cover, distance to co-graph, distance to interval graph, pathwidth, cluster vertex deletion number, $W[2]$-hard with respect to the parameter seed set cardinality, and fixed parameter tractable with respect to the parameters distance to clique and bandwidth. Dvořák et al. [8] added a few more results in the parameterized setting. They showed that TSS is $W[1]$-hard with respect to parameter neighborhood diversity, and under majority threshold (i.e., the threshold of each vertex is equal to half of its degree) this problem has an FPT algorithm with respect to the parameters neighborhood diversity and twin cover, and $W[1]$-hard with respect to the parameter modular width. Bazgan et al. [9] showed that for any functions $f$ and $\rho$ this problem cannot be approximated within a factor of $\rho(k)$ in $f(k) \cdot n^{O(1)}$ time unless $\text{FPT} = \text{W}[P]$ even for constant and majority thresholds, where $k$ denotes the cardinality of an optimal target set. Nichterlein et al. [10] showed that for diameter two split graphs TSS remains $W[2]$-hard with respect to the parameter size of the target set. Also, TSS is fixed parameter tractable when parameterized by the vertex cover number and cluster editing number. Hartmann [11] showed that the TSS Problem is FPT when parameterized with the combined parameters clique-width and maximum threshold value of the input graph. Notice that clique-width is smaller than many parameters like vertex cover number and treewidth. Bliznets et al. [12] presented several faster-than-trivial algorithms under several threshold models such as constant thresholds, dual constant thresholds where the threshold value of each vertex is bounded by one third of its degree. Recently, Keiler et al. [13] studied a variant of TSS Problem where the goal is to select a target set that maximizes the diffusion rounds. They showed that the problem is FPT when parameterized by diffusion rounds and maximum threshold if the input graph has bounded local treewidth. The problem is NP-Complete even when we need the diffusion rounds to be at least 4 and maximum threshold is equal to 2. They also showed that the problem is NP-Hard for planar graphs, $W[1]$-Hard parameterized by treewidth, and polynomial-time solvable on trees.
Our Contribution For any non-negative integer \( i \), we say that a set \( T \subseteq V(G) \) is a degree-\( i \) modulator of \( G \) if the degree of any vertex in the graph \( G - T \) is at most \( i \). Degree-0 modulators of a graph are precisely its vertex covers. Consider a graph \( G \) on \( n \) vertices and \( m \) edges. We have the following results on TSS.

- Nichterlein et al. [Social Network Analysis and Mining, 2013] showed that it is possible to compute an optimal-sized target set in \( O(2^{(2^t+1)t} \cdot m) \) time, where \( t \) denotes the cardinality of a minimum degree-0 modulator (i.e., vertex cover) of \( G \). In Section 3, we improve this result by designing an algorithm that computes an optimal-sized target set in \( 2^{O(t \log t)} n \) time.
- We introduced a new problem called MULTI-HITTING SET which may be of independent interest. The input for this problem is a universe \( U \) of size \( n \), a collection of subsets \( S_1, S_2, \ldots, S_t \subseteq U \) and \( q, \ell_1, \ell_2, \ldots, \ell_t \in \mathbb{N} \) such that \( \ell_j \leq t \) for all \( j \in [t] \). The objective is to test whether there exists a subset \( H \subseteq U \), such that \( |H| \leq q \) and \( |H \cap S_i| \geq \ell_i \) for all \( i \in [t] \). In Section 3, we prove that MULTI-HITTING SET can be solved in time \( 2^{O(t \log t)} n \) time. To prove that TSS parameterized by the vertex cover number \( t \) is solvable in time \( 2^{O(t \log t)} n \), we give a Turing reduction to MULTI-HITTING SET.
- In Section 4, we design a \( 2^{O(t)} n^{O(1)} \) time algorithm to compute an optimal target set, where \( t \) is the size of a minimum degree-1 modulator of \( G \). (We would like to mention that Knop et al. [14] independently proved that the TSS problem can be solved in time \( t^{O(t^4)} n^{O(1)} \) using \( N \)-fold Integer Programming, where \( t \) is the size of a minimum degree-1 modulator of \( G \). Our work [15] was submitted to ArXiv on 16-May-2021 while Knop et al. [14] submitted their work to ArXiv on 21-May-2021.)
- Ben-Zwi et al. [16] proved algorithmic lower bounds for TSS when parameterized by the treewidth of the input graph. Towards that, in Lemma 4.3 of [16], the authors give a polynomial-time reduction that maps an instance \((G, k)\) of the MULTI-COLORED CLIQUE problem\(^1\) to an instance \((G', \tau, k')\) of the TARGET SET SELECTION (TSS) problem, where \( k' = k + \left(\frac{k}{2}\right) \), such that \( G \) has a multicolored clique of size \( k \) if and only if \( G' \) has a target set of size \( k' \). The star-deletion number (minimum number of vertices whose removal results in a graph that is a disjoint collection of star graphs) of the constructed graph \( G' \) is \( O(k^2) \). Thus, the treewidth and tree-depth of \( G' \) is \( O(k^2) \). Ben-Zwi et al. [16] show that the MULTI-COLORED CLIQUE problem cannot be solved in \( n^{\omega(k)} \) time unless all problems in SNP can be solved in sub-exponential time. This helps them in proving that for a graph on \( n \) vertices of treewidth (or tree-depth, or star-deletion number) \( s \), the TSS problem cannot be solved in \( f(s)n^{\omega(\sqrt{s})} \) time unless the Exponential Time Hypothesis (ETH) fails. For definitions of treewidth and tree-depth, see [17].

\(^1\)Given a set of \( k \) distinct colors and a graph on \( n \) vertices whose every vertex is colored with one of the \( k \) colors, the MULTI-COLORED CLIQUE problem is about finding a clique of size \( k \) in the graph such that no two vertices of the clique are of the same color.
It was shown in [18, Corollary 6.3] that if the Partitioned Subgraph Isomorphism (PSI) problem\(^2\) can be solved in \(f(G)n^{o(\frac{1}{\log k})}\) time, then ETH fails, where \(f\) is an arbitrary function, \(n = |V(H)|\), \(G\) is connected, and \(k\) is the number of edges of the smaller graph \(G\). We observe that by making small modifications to the reduction given in [16], we can give a polynomial-time reduction from an instance \((G, H, f_G, C_H)\) of the PSI problem, where \(|V(G)| = \ell, |E(G)| = k,\) and \(|V(H)| = n\), to an instance \((G', \tau, k')\) of the TSS problem, where \(k' = \ell + k\), such that \((G, H, f_G, C_H)\) is a YES-instance of the PSI problem if and only if \(G'\) has a target set of size \(k'\). The star-deletion number of the graph \(G'\) constructed can be shown to be \(O(k)\). This helps us in improving the lower bound given in [16]. We can show that for a graph on \(n\) vertices of treewidth (or tree-depth, or star-deletion number) \(s\), the Target Set Selection problem cannot be solved in \(f(s)n^{o(\frac{s}{\log s})}\) time unless ETH is false. The reduction is similar to the one in [16], where we use PSI instead of Multi-Colored Clique. For details, see Section 5.

1.1 Preliminaries

Throughout the paper, we consider finite, undirected and simple graphs. For any vertex \(v\) in a graph \(G\), we shall use \(d_G(v)\) to denote the number of edges incident on \(v\) and \(N_G(v)\) to denote the set of vertices adjacent to \(v\). We omit the subscript \(G\) if the graph \(G\) is clear from the context. For any subset, say \(S\), of the set of vertices of the graph under consideration, we shall use \(N_S(v)\) to denote the set of neighbors of \(v\) in \(S\). For a graph \(G\), we shall use \(V(G)\) and \(E(G)\) to denote its vertex set and edge set, respectively. For a set \(S \subseteq V(G)\), we shall use \(G[S]\) to denote the subgraph induced by \(S\) on \(G\), and \(G - S\) to denote the graph \(G[V(G) \setminus S]\). A set \(S \subseteq V(G)\) is said to be an independent set of \(G\) if no two vertices in \(S\) are adjacent with each other. A set \(U \subseteq V(G)\) is a vertex cover of \(G\), if \(V(G) \setminus U\) is an independent set. A vertex cover with minimum cardinality is called an optimal vertex cover. For a function \(f : X \mapsto Y\), \(X' \subseteq X\) and \(Y' \subseteq Y\), \(f(X') = \{ f(x) : x \in X' \}\) and \(f^{-1}(Y') = \{ x \in X : f(x) \in Y' \}\).

2 Reduction Rules, Auxiliary Lemmas, and Overview of our Algorithms

In this section we state some simple reduction rules which we apply to any instance if applicable. Then, we state some auxiliary lemmas and give a brief overview of our algorithms. TSS when parameterized by degree-\(i\) modulator is formally defined below.

\(^2\)In PSI we are given two graphs \(G\) and \(H\), a bijection \(f_G : V(G) \mapsto [\ell]\), and a function \(c_H : V(H) \mapsto [\ell]\), where \(|V(G)| = \ell\). The objective is to test the existence of a subgraph isomorphism \(\phi\) from \(G\) to \(H\) such that for all \(v \in V(G)\), \(f_G(v) = c_H(\phi(v))\).

Springer
Recall that TSS-DOM is the TSS problem parameterized by vertex cover. First we define couple of simple reduction rules. A reduction rule for a parameterized problem $\Pi$ is an algorithm that takes an instance $(x, k)$ of $\Pi$ and outputs an instance $(y, k')$ such that $(x, k)$ is YES-instance of $\Pi$ if and only if $(y, k')$ is a YES-instance of $\Pi$.

**Observation 1** [10] Let $G$ be a graph with a threshold function $\tau : V(G) \to \mathbb{N}$. Let $v \in V(G)$ such that $\tau(v) > d_G(v)$. Then, $v$ is contained in all target sets for $G$.

The correctness of the following reduction rule follows from Observation 1.

**Reduction Rule 1** [10] Let $(G, \tau, k, C)$ be an instance of TSS-DiM. Let $v \in V(G)$ such that $\tau(v) > d_G(v)$. Then, delete $v$, decrease the threshold of all its neighbors by one and decrease $k$ by one.

**Observation 2** Let $G$ be a graph with a threshold function $\tau : V(G) \to \mathbb{N}$. Let $v \in V(G)$ such that $\tau(v) \leq d_G(v) \leq 1$. Then, there is an optimal target set that does not contain $v$.

**Proof** Let $S$ be an optimal target set for $G$ such that $v \in S$. Then, $S' = S \cup N(v)$ is a target set of size at most $|S| + 1$. Since $\tau(v) \leq d_G(v)$ and $N(v) \subseteq S'$, $S' \setminus \{v\}$ is a target set for $G$ and $|S' \setminus \{v\}| \leq |S|$.

**Reduction Rule 2** Let $(G, \tau, k, C)$ be an instance of TSS-DiM. Let $v \in V(G)$ such that $\tau(v) \leq d_G(v) \leq 1$. Then, then delete $v$.

Next we state a simple observation that a vertex cover is a solution if the instance is reduced with respect to Reduction Rule 1.

**Observation 3** Let $G$ be a graph with a threshold function $\tau : V(G) \to \mathbb{N}$ such that for all $v \in V(G)$, $\tau(v) \leq d_G(v)$. Let $C$ be a vertex cover of $G$. Then, $C$ is a target set for $G$.

Next we prove two lemmas regarding the number of rounds in the diffusion process in terms of the cardinality of a vertex cover or degree-1 modulator.

**Lemma 4** Let $t$ be the size of an optimal vertex cover in a graph $G$ with a threshold function $\tau : V(G) \to \mathbb{N}$. Then, the diffusion process starting from any non-empty seed set $S \subseteq V$ terminates in at most $2t$ rounds.
Proof Let $C$ be an optimal vertex cover of $G$ of size $t$ and let $B = V(G) \setminus C$. We know that $B$ is an independent set. Let $S_i$ be the set of uninfluenced nodes that were influenced in Round $i$ of the diffusion process. We have $S_0 = S$. Assume the diffusion process terminates in $k$ rounds. For each $0 \leq i \leq k$, observe that $S_i$ is a non-empty set. For $0 \leq i < k$, since $B$ is an independent set, it is not possible to have both $S_i$ and $S_{i+1}$ to be subsets of $B$. Thus, in every two consecutive rounds, at least one uninfluenced vertex from $C$ will be influenced. This implies that $C \cap \text{influence}_G(S)$ will be influenced in at most $2t - 1$ steps. Therefore, the diffusion process will terminate in at most $2t$ steps.

Lemma 5 Let $t$ be the size of an optimal degree-1 modulator in a graph $G$ with a threshold function $\tau: V(G) \to \mathbb{N}$. Then, the ‘diffusion process’ starting from any non-empty seed set $S \subseteq V$ terminates in at most $3t$ rounds.

Proof Let $C$ be an optimal degree-1 modulator of $G$ of size $t$ and let $B = V(G) \setminus C$. Notice that $G[B]$ is a collection of isolated vertices and isolated edges. Let $S_i$ be the set of uninfluenced nodes that were influenced in Round $i$ of the diffusion process. We have $S_0 = S$. Assume the diffusion process terminates in $\ell$ rounds. For each $0 \leq i \leq \ell$, observe that $S_i$ is a non-empty set. For $0 \leq i \leq \ell - 2$, since $G[B]$ is a collection of isolated vertices and isolated edges, it is not possible to have $S_i$, $S_{i+1}$, and $S_{i+2}$ to be subsets of $B$. Thus, in every three consecutive rounds, at least one uninfluenced vertex from $C$ will be influenced. This implies that $C \cap \text{influence}_G(S)$ will be influenced in at most $3t - 1$ steps. Therefore, the diffusion process will terminate in at most $3t$ steps. This proves the lemma.

Overview of our Algorithms Now we give a brief overview of our algorithms for TSS-D0M and TSS-D1M. Initially we apply Reduction Rules 1 and 2 as long as it is applicable. Then, for the reduced instance $(G, \tau, k, C)$ we have that (i) for any $v \in V(G)$, $\tau(v) \leq d_G(v)$ and $d_G(v) \geq 2$. By Lemmas 4 and 5, we know that the diffusion process terminates in at most $3t$ steps. Thus, as a first step we guess the time stamp at which the vertices in the modulator get activated by a hypothetical solution. Then, since the size of an optimal target set in the case of TSS-D0M is at most the size of the modulator (see Observation 3), we reduce the problem to a new problem called MULTI-HITTING SET where the the hitting set will serve as the vertices in the target set from $V(G) \setminus C$. But, the size of an optimal solution even for a reduced instance of TSS-D1M may not be bounded by a function of the modulator. For example, suppose $G$ is a collection of $n'$ isolated edges together with a universal vertex $v$. Thus, $G'$ is a graph on $2n' + 1$ vertices where every vertex other than $v$ has degree 2. Let the threshold for all the vertices be their degrees. Note that neither Reduction Rule 1 nor Reduction Rule 2 applies to $G$. The size of any target set for $G$ is at least $n' + 1$, whereas $\{v\}$ is a degree-1 modulator of $G$ of size 1. Because of this, for TSS-D1M, we encode it as an ILP with number of variables bounded by a function of the size of the modulator.
3 FPT Algorithm Parameterized by Degree-0 Modulator (Vertex Cover Number)

Recall that TSS-D0M is the TSS problem parameterized by vertex cover. Throughout this section we assume that the instance is reduced with respect to Reduction Rules 1 and 2. Toward getting our FPT algorithm, we give a Turing reduction from TSS-D0M to a variant of the hitting set problem which is defined below.

Now, we describe our Turing reduction that constructs $2^{O(t \log t)}$ instances of MULTİ-HİTTİNG SET from a given instance of TSS-D0M.

**Theorem 6** There is an algorithm that given an instance $(G, \tau, k, C)$ of TSS-D0M, where $t = |C|$, runs in $2^{O(t \log t)} \cdot n$ time, and outputs a collection of instances $I = \{I^j = (V(G) \setminus C, q^j, S^j_1, S^j_2, \ldots, S^j_t, \ell^j_1, \ell^j_2, \ldots, \ell^j_t) : j \in [s]\}$ of MULTİ-HİTTİNG SET such that the following holds:

(a) The number of instances, i.e., $s \leq 2^{O(t \log t)}$.

(b) $(G, \tau, k, C)$ is a YES-instance of TSS-D0M if and only if there exists $j \in [s]$ such that $I^j$ is a YES-instance of MULTİ-HİTTİNG SET.

**Proof** First we describe the construction of $2^{O(t \log t)}$ many instances of MULTİ-HİTTİNG SET from a given instance $(G, \tau, k, C)$ of TSS-D0M.

**Construction** Consider the given TSS-D0M instance $(G, \tau, k, C)$, where $G$ is a graph on $n$ vertices and $C = \{v_1, \ldots, v_t\}$ is a vertex cover of size $t$. Let $B = V(G) \setminus C$. Note that $B$ is an independent set. For each $v_i \in C$, we guess a time stamp $T(v_i)$ in which $v_i$ will be influenced. From Lemma 4, we know that $T(v_i) \in \{0, \ldots, 2t\}$. There are $t$ vertices in $C$ and each one of them can be assigned any one of these $2t + 1$ distinct values. So, there are $(2t + 1)^t = 2^{O(t \log t)}$ possible guesses for the time stamps of the vertices in $C$. Now, among all the $2^{O(t \log t)}$ possibilities, let us consider the $j$-th one. That is, consider that we are given a $t$-tuple $(T^j(v_1), T^j(v_2), \ldots, T^j(v_t))$ of guessed time stamps for vertices in $C$. Based on these guessed time stamps for the vertices in $C$, for any $u \in B$, we compute $T^j(u)$ as $1 + \min\{x \in \mathbb{N} : \text{number of vertices in } N(u) \text{ with time stamp at most } x, \text{ is at least } \tau(u)\}$. For each $v_i \in C$, we define $\ell^j_i := \max\{0, \tau(v_i) - |\{w \in N(v_i) : T^j(w) < T^j(v_i)\}|\}$ and $S^j_i := \{w \in N_B(v_i) : T^j(w) \geq T^j(v_i)\}$. We remark that if $\ell^j_i \geq t$, then we will not include $I^j$ in the collection of the output instances. We thus have the $j$-th instance of the multi-hitting set problem where $U = B$, $S^j_i$’s and $\ell^j_i$’s are as defined above, $t = |C|$, and $q^j = k - |\{v_i \in C : T(v_i) = 0\}|$. This completes the construction of
output instances. It is easy to verify that the number of instance in $\mathcal{I}$ is $2^{O(t \log t)}$. Property (b) follows from the following two claims.

Claim 1 Suppose it is given that one of the $t$-tuples we guess, say the $j$-th $t$-tuple $(T^j(v_1), T^j(v_2), \ldots, T^j(v_t))$, happens to represent the activation time of vertices in $C$ corresponding to some feasible target set $S$. Then, for all $i \in [t]$, $|S \cap S_i^j| \geq \ell_i^j$. That is, $S \cap B$ is a solution for the instance $I^j$ of MULTI-HITTING SET.

Proof For any $w \in B$, $N(w) \subseteq C$ and we know that for any $v_i \in C$, $v_i$ is influenced in step $T^j(v_i)$ for the target set $S$. This implies that for any $w \in B \setminus S$, $w$ is influenced in step $T^j(w)$. Therefore, as $v_i$ is influenced in step $T^j(v_i)$, at least $\ell_i^j$ vertices from $S^j_i$ should be there in the target set $S$. This implies that $S \cap B$ is a solution for the instance $I^j$. \qed

Claim 2 Let $H$ be a hitting set for the instance $I^j$ of MULTI-HITTING SET. Then $S = H \cup \{v_i \in C : T^j(v_i) = 0\}$ is a target set for $G$.

Proof To prove the claim it is enough to show that $C \subseteq \text{influence}_G(S)$. We prove by induction on $r$ that all the vertices $v_i \in C$ with $T^j(v_i) \leq r$ will be influenced by the end of step $r$. The base case is when $r \leq 1$. Clearly, for $r = 0$, all the vertices $v_i \in C$ with $T^j(v_i) \leq 0$ is influenced initially because $v_i \in S$. Notice that for any $w \in B$, $T^j(w) > 0$ and since $H$ is a solution for the instance $I^j$ of MULTI-HITTING SET, for any $v_i \in C$ with $T^j(v_i) = 1$, at least $\ell_i^j = \tau(v_i)$ vertices from $N(v_i)$ are there in $H$. This implies that for any $v_i \in C$ with $T^j(v_i) = 1$, $v_i$ will be influenced in step 1. Now consider the induction step for which $r > 1$. Now consider a vertex $v_i \in C$ with $T^j(v_i) = r$. We know that at least $\ell_i^j$ vertices from $S^j_i := \{w \in N_B(v_i) : T^j(w) \geq r\}$ are present in $H$ because $H$ is a solution to $I^j$. By the induction hypothesis, we have that for any vertex $v_a \in C$ with $T^j(v_a) = r - 2$ is influenced at the end of step $r - 2$. This implies that all the vertices in $\{w \in N_B(v_i) : T^j(w) < r\}$ are influenced by the end of step $r - 1$. Therefore, at least $\tau(v_i)$ vertices from $N(v_i)$ will be influenced by the end of step $r - 1$. This implies that $v_i$ will be influenced in step $r$. This completes the proof of the claim. \qed

This completes the proof of the theorem.

Next, we design an FPT algorithm for MULTI-HITTING SET.

Theorem 7 Given an instance $I = (U, q, S_1, S_2, \ldots, S_t, \ell_1, \ell_2, \ldots, \ell_t)$ of MULTI-HITTING SET, there is an algorithm of running time $2^{O(t \log t)} \cdot n$ to solve $I$, where $|U| = n$.

Proof Let $U = \{u_1, \ldots, u_n\}$ and let $U_j = \{u_1, \ldots, u_j\}$, $1 \leq j \leq n$. We design a dynamic programming algorithm, where in the DP table entry $D_j(q', \ell_1', \ell_2', \ldots, \ell_t')$ we store a hitting set (if one exists; else, it will be equal to NULL) of size at most $q'$ that is a subset of $U_j$ and hits each $S_i$ on at least $\ell_i'$ elements, where $0 \leq \ell_i' \leq \ell_i$, for all $i \in [t]$ and $0 \leq q' \leq q$. The case when $j = 0$ can be computed easily as follows.
\[
D_0(q', \ell'_1, \ell'_2, \ldots, \ell'_t) = \begin{cases} 
\emptyset & \text{if } \ell'_i = 0 \text{ for all } i, \\
\text{NULL} & \text{otherwise.}
\end{cases}
\]

We compute the DP table entries in increasing order of \( j \). Consider the case when \( j \geq 1 \). Without loss of generality, assume \( u_j \in S_1 \cap \cdots \cap S_k \) and \( u_j \notin S_{k+1} \cup \cdots \cup S_t \). Then, for any values of \( q', \ell'_1, \ell'_2, \ldots, \ell'_t \) such that \( 0 \leq \ell'_i \leq \ell_i \), for all \( i \in [t] \) and \( 0 \leq q' \leq q \), we compute \( D_j(q', \ell'_1, \ell'_2, \ldots, \ell'_t) \) as follows. If \( D_{j-1}(q', \ell'_1, \ell'_2, \ldots, \ell'_t) \neq \text{NULL}, \) then
\[
D_j(q', \ell'_1, \ell'_2, \ldots, \ell'_t) = D_{j-1}(q', \ell'_1, \ell'_2, \ldots, \ell'_t).
\]
If \( D_{j-1}(q' - 1, \ell'_1 - 1, \ell'_2 - 1, \ldots, \ell'_k - 1, \ell'_{k+1}, \ldots, \ell'_t) \neq \text{NULL} \), then set
\[
D_j(q', \ell'_1, \ell'_2, \ldots, \ell'_t) = D_{j-1}(q' - 1, \ell'_1 - 1, \ell'_2 - 1, \ldots, \ell'_k - 1, \ell'_{k+1}, \ldots, \ell'_t) \cup \{u_j\}.
\]
Otherwise, we set \( D_j(q', \ell'_1, \ell'_2, \ldots, \ell'_t) = \text{NULL} \). Using this dynamic programming approach, we eventually compute \( D_n(q, \ell_1, \ldots, \ell_t) \). Since each of \( \ell_1, \ldots, \ell_t \) is at most \( t \) and the cardinality of an optimum solution is at most \( t^2 \) (and hence we assume that \( q \leq t^2 \)), we can compute this in time \((t+1)^tn = 2^{O(t \log t)} \cdot n\).

Next we prove the correctness of our algorithm. Towards that we claim that \( D_n(q, \ell_1, \ldots, \ell_t) = \text{NULL} \) if and only if \( I \) is a NO-instance. We prove this by showing a more general statement. We will show that, for every \( j \in \{0, \ldots, n\} \), \( D_j(q', \ell'_1, \ldots, \ell'_t) = \text{NULL} \) if and only if \( I_j(q'_1, \ell'_1, \ldots, \ell'_t) := (U_j, q'_1, S_{j, 1}, \ldots, S_{j, t}; \ell'_1, \ldots, \ell'_t) \) is a NO-instance where \( S_{i, j} = S_i \cap U_j, \ell'_i \leq \ell_i \), for every \( i \), and \( 0 \leq q' \leq q \). We prove this by strong induction on \( j \). It is easy to see that the statement is true for the base case when \( j = 0 \). Consider the induction step when \( j > 0 \). Notice that, by the induction hypothesis, the statement is true for all \( j' < j \). Without loss of generality, assume \( u_j \in S_1 \cap \cdots \cap S_k \) and \( u_j \notin S_{k+1} \cup \cdots \cup S_t \). Suppose \( D_j(q', \ell'_1, \ldots, \ell'_t) = \text{NULL} \). This implies, both \( D_{j-1}(q' - 1, \ell'_1 - 1, \ldots, \ell'_k - 1, \ell'_{k+1}, \ldots, \ell'_t) \) and \( D_{j-1}(q' - 1, \ell'_1 - 1, \ell'_2 - 1, \ldots, \ell'_k - 1, \ell'_{k+1}, \ldots, \ell'_t) \) are NO-instances. Hence, \( I_j(q'_1, \ell'_1, \ldots, \ell'_t) \) is a NO-instance.

Now, to prove the reverse direction of the bidirectional statement, assume that \( I_j(q'_1, \ell'_1, \ldots, \ell'_t) \) is a NO-instance. Then, clearly \((U_{j-1}, q'_1 - 1, S_{i, j-1}, \ldots, S_{i, t}; \ell'_{j-1}' - 1, \ldots, \ell'_{k-1}, \ldots, \ell'_t)\) and \((U_{j-1}, q'_1, S_{i, j-1}, \ldots, S_{i, t}; \ell'_{j-1}' - 1, \ldots, \ell'_t)\) are NO-instances, where \( S_{i, j-1} = S_i \cap U_{j-1} \) for every \( 1 \leq i \leq t \). Therefore, both \( D_{j-1}(q' - 1, \ell'_1 - 1, \ell'_2 - 1, \ldots, \ell'_k - 1, \ell'_{k+1}, \ldots, \ell'_t) = \text{NULL} \) and \( D_{j-1}(q', \ell'_1, \ldots, \ell'_t) = \text{NULL} \). Hence, \( D_j(q', \ell'_1, \ldots, \ell'_t) = \text{NULL} \). This proves the theorem. \( \square \)

Below we state the main result of this section which follows directly from Theorems 6 and 7 and from the fact that a linear time factor-2 approximation algorithm exists for computing a minimum vertex cover in a graph because the set of endpoints of an inclusion-wise maximal matching is a factor 2-approximate solution for vertex cover.
Theorem 8 Let $G$ be a graph on $n$ vertices with a threshold function $\tau : V(G) \to \mathbb{N}$ defined on its vertices. Let $t$ be the size of an optimal vertex cover in $G$. Then, the optimal target set for $G$ can be computed in time $2^{O(t \log t)}n$.

4 FPT Algorithm Parameterized by Degree-1 Modulator

In this section we prove the following theorem.

Theorem 9 TSS-D1M is solvable in time $2^{2^{O(t)}}n^{O(1)}$.

As there is a simple branching algorithm of running time $O(3^t (n + m))$ to compute a degree-1 modulator, as a consequence of Theorem 9, given a graph $G$ with a threshold function $\tau$, we can obtain an optimum target set in time $2^{2^{O(t)}}n^{O(1)}$, where $t$ is the size of a minimum degree-1 modulator of $G$.

As before we assume that the given instance is reduced with respect to Reduction Rules 1 and 2. That is, the minimum degree of a vertex in $G$ is at least 2 and for any vertex $v \in V(G)$, $\tau(v) \leq d_G(v)$. Recall that for a degree-1 modulator $S$ in a graph $G$, $G - S$ is a disjoint union of isolated vertices and isolated edges.

We encode TSS-D1M as an INTEGER PROGRAMMING (IP) problem. In IP, we are given $m$ linear constraints over $n$ variables and we want to check whether there is an integer assignment to the variables such that all the constraints are satisfied. More formally, the input consists of an $m \times n$ matrix $A$ and a $m$-length column vector $b$, and the objective is to test whether there exists an $n$-length vector $x$ with all integer coordinates such that $Ax \leq b$.

By the famous result of Lenstra [19], we know that IP parameterized by the number of variables is FPT. The current best known running time for solving an IP with $n$ variables and $m$ constraints is $2^{O(n \log n)}m^{O(1)}$ [20].

Proposition 10 [20] IP is solvable in time $2^{O(n \log n)}L \log L$, where $n$ is the number of variables and $L$ is the input length.

We reduce TSS-D1M to many instance of IP such that the number of variables in each instances of IP is bounded by a function of $t$ and TSS-D1M is a YES-instance if and only if at least one of the instances of IP is a YES-instance. Before giving a Turing reduction from TSS-D1M to IP, we prove some results which we use in the reduction.

Observation 11 Let $G$ be a graph and $\tau : V(G) \to \mathbb{N}$ be a threshold function. Let $\{u, v\}$ be an edge in the graph $G$ such that $\tau(u) = d_G(u)$ and $\tau(v) = d_G(v)$. Then, for any target set, at least one vertex from $\{u, v\}$ belongs to $S$.

Proof Consider the set $Q = V(G) \setminus \{u, v\}$. Since $\tau(u) = d_G(u)$ and $\tau(v) = d_G(v)$, both $u$ and $v$ will get influenced only after all of their neighbors get influenced. This
implies that \( Q \) is not a target set. Hence any subset of \( Q \) is not a target set as well. Thus, at least one vertex from \( \{u, v\} \) should belongs to \( S \) for any target set \( S \).

**Lemma 12** There is an algorithm that given a graph \( G \), a threshold function, \( \tau: V(G) \to \mathbb{N} \), a degree-1 modulator \( C \) and a time stamp \( T: C \to \mathbb{N} \cup \{0\} \) of an unknown target set \( S \) (a hypothetical solution), runs in linear time, and outputs two functions \( g: V(G) \to \mathbb{N} \) and \( h: V(G) \to \mathbb{N} \) with the following properties. Let \( u \in V(G) \setminus (C \cup S) \) and if \( d_{G-C}(u) = 1 \), then \( v \) is the only neighbor of \( u \) in \( G - C \). (Notice that \( d_{G-C}(u) \leq 1 \)).

(i) If \( d_{G-C}(u) = 0 \), then \( g(u) = h(u) \) and \( u \) is influenced on step \( g(u) \) in the diffusion process starting from the target set \( S \).

(ii) If \( d_{G-C}(u) = 1 \) and \( v \in S \), then \( g(u) \leq h(u) \) and \( u \) is influenced on step \( g(u) \) in the diffusion process starting from the target set \( S \).

(iii) If \( d_{G-C}(u) = 1 \) and \( v \notin S \), then \( g(u) \leq h(u) \) and \( u \) is influenced on step \( h(u) \) in the diffusion process starting from the target set \( S \).

**Proof** Recall that the minimum degree of a vertex in \( G \) is at least 2. Notice that we are given a time stamp \( T: C \to \mathbb{N} \cup \{0\} \). Let \( B = V(G) \setminus C \). First we define the function \( g \) as follows. Fix a vertex \( u \in B \). Suppose \( d_{G-C}(u) = 0 \). Then, let \( g(u) \) be the smallest positive integer \( i \) such that \( | \bigcup_{j<i}(T^{-1}(j) \cap N_G(u)) | \geq \tau(u) \). Now suppose \( d_{G-C}(u) = 1 \). Then, let \( g(u) \) be the smallest positive integer \( i \) such that \( | \bigcup_{j<i}(T^{-1}(j) \cap N_G(u)) | \geq \tau(u) - 1 \). It is easy to see that \( g \) is well defined.

Next we define the function \( h \). For a vertex \( u \in B \), if \( d_{G-C}(u) = 0 \), then \( h(u) = g(u) \). Now for all vertices \( u \) such that \( d_{G-C}(u) = 1 \), we define \( h \) iteratively such that the set \( h^{-1}(i) \) is defined after defining \( h^{-1}(j) \) for all \( j < i \). First we explain the vertices that will be mapped to 1 by the function \( h \). For each vertex \( u \in B \) with \( d_{G-C}(u) = 1 \), if \( | T^{-1}(0) \cap N_G(u) | \geq \tau(u) \), then we define \( h(u) = 1 \). Now let us consider an integer \( i > 1 \) and assume that we have defined \( h^{-1}(j) \) for all \( j < i \). Let \( Y_{<i} = \bigcup_{j<i}(T^{-1}(j) \cup h^{-1}(j)) \). Let \( u \) be a vertex in \( B \setminus Y_{<i} \) such that \( d_{G-C}(u) = 1 \). We set \( h(u) = i \) if \( | Y_{<i} \cap N_G(u) | \geq \tau(u) \). If there is a vertex \( u \) such that \( h(u) \) is not defined so far, then we set \( h(u) = \infty \). Clearly, \( h \) is well defined. Since for any vertex \( u \in B \), \( d_{G-C}(u) \leq 1 \), by the definition of \( g \) and \( h \), we have that \( g(u) \leq h(u) \). Next, we prove that the properties (i)-(iii) are satisfied.

**Proof of property (i)** Here, we prove that the function \( g \) and \( h \) satisfies the property (i) mentioned in the lemma. Recall that \( S \) is a target set and we want to prove that for all vertex \( u \in B \setminus S \) with \( d_{G-C} = 0 \), \( g(u) = h(u) \) and \( u \) is influenced on step \( g(u) \) in the diffusion process starting from \( S \). Since \( T \) is the time stamp of \( S \) on \( C \) and \( N_G(u) \subseteq C \), \( u \) is influenced on step \( i \), where \( i \) is the least integer such that \( | \bigcup_{j<i}(T^{-1}(j) \cap N_G(u)) | \geq \tau(u) \). By the definition of \( g \) and \( h \), we have that \( g(u) = i \) and \( h(u) = g(u) \).

**Proof of property (ii)** Let \( u \in B \setminus S \) with \( d_{G-C} = 1 \) and \( v \) be the only neighbor of \( u \) in \( G - C \). By our assumption, we have that \( v \in S \). Thus, \( u \) is influenced on step \( i \), where \( i \) is the least integer such that \( | \bigcup_{j<i}(T^{-1}(j) \cap N_G(u)) | \geq \tau(u) - 1 \). By the definition of \( g \), we have that \( g(u) = i \).
Proof of property (iii) We prove using induction on \( i \) that any vertex \( u \in B \setminus S \) such that \( d_{G-C}(u) = 1 \) and \( v \notin S \) (where \( v \) is the only neighbor of \( u \) in \( G-C \)), \( u \) is influenced on step \( i \) if and only if \( h(u) = i \). The base case is when \( i = 1 \). Let \( u \) be a vertex in \( B \setminus S \) such that \( h(u) = 1, d_{G-C}(u) = 1, \) and \( v \notin S \) (where \( v \) is the only neighbor of \( u \) in \( G-C \)). Suppose, \( h(u) = 1 \). By the definition of \( h \), we have that \(|T^{-1}(0) \cap N_G(u)| \geq \tau(u)\). This implies that \( u \) is influenced on step 1. Now, for the other direction, suppose \( u \) is influenced on step 1. Thus, since \( v \notin S \), we have that \(|T^{-1}(0) \cap N_G(u)| \geq \tau(u)\). Therefore, by the definition of \( h \), \( h(u) = 1 \).

Now consider the induction step \( i > 1 \). Let \( u \) be a vertex in \( B \setminus S \) such that \( d_{G-C}(u) = 1 \), \( v \notin S \) (where \( v \) is the only neighbor of \( u \) in \( G-C \)).

(\( \Rightarrow \)) Suppose \( h(u) = i \). Let \( j \) be the integer such that \( v \) is influenced on step \( j \) in the diffusion process starting from \( S \). Consider the case when \( j < h(u) \). Since \( d_{G-C}(v) = 1, v \) and \( u \) do not belong to \( S \), by induction hypothesis we have that \( h(v) = j \). Since \( h(u) = i \), \( i \) is the least integer such that \(|Y_{<i} \cap N_G(u)| \geq \tau(u)\) where \( \{v\} = (Y_{<i} \setminus C) \cap N_G(u) \). Since \( T \) is a the time stamp on \( C \) and \( h(v) = j \), \( u \) is influenced on step \( i \). Now consider the case when \( j \geq h(u) \). Then, by induction hypothesis, \( h(v) \geq j \). This implies that \(|Y_{<i} \cap N_G(u)| = \bigcup_{j<i}(T^{-1}(j) \cap N_G(u))\) because \( v \notin Y_{<i} \). Also, since \( i \) is the least integer such that \(|Y_{<i} \cap N_G(u)| \geq \tau(u)\), we have that \( u \) is influenced on step \( i \).

(\( \Leftarrow \)) Suppose \( i \) be the integer such that \( u \) is influenced on step \( i \). Let \( j = h(v) \). Consider the case when \( j < h(u) \). By induction hypothesis \( v \) is influenced on step \( j \). Therefore, \( i \) is the least integer such that \(|Y_{<i} \cap N_G(u)| \geq \tau(u)\) where \( \{v\} = (Y_{<i} \setminus C) \cap N_G(u) \). Hence, \( h(u) = i \). Now consider the case when \( j \geq h(u) \). By induction hypothesis \( v \) is not influenced on or before step \( h(u) - 1 \). Therefore \( i \) is the least integer such that \(|\bigcup_{j<i}(T^{-1}(j)) \cap N_G(u)| \geq \tau(u)\). Since \( v \notin Y_{<i} \), \(|Y_{<i} \cap N_G(u)| = |\bigcup_{j<i}(T^{-1}(j)) \cap N_G(u)| \geq \tau(u)\). Hence, \( h(u) = i \).

Theorem 13 There is an algorithm that given an instance \((G, \tau, k, C)\) of TSS-D1M, where \( t = |C| \), runs in \( 2^{O(t \log t)} \cdot n^{O(1)} \) time, and outputs a collection of instances \( \mathcal{I} \) of IP such that the following holds:

(a) The number of instances is \( 2^{O(t \log t)} \).
(b) For every \( I \in \mathcal{I} \), the number of variables in \( I \) is upper bounded by \( 2^{O(t)} \).
(c) \((G, \tau, k, C)\) is a YES-instance of TSS-D1M if and only if there exists \( I \in \mathcal{I} \) such that \( I \) is a YES-instance of IP.

Proof Consider the given instance \((G, \tau, k, C)\) TSS-D1M, where \( G \) is a graph on \( n \) vertices and \( C := \{v_1, \ldots, v_t\} \) is a degree-1 modulator of \( G \) of size \( t \). Without loss of generality we assume that for each vertex \( v \in V(G) \), \( \tau(v) \leq d_G(v) \). Let \( B = V(G) \setminus C \). We know that the maximum degree in \( G[B] \) is at most 1. That is \( G[B] \) is a disjoint union of isolated vertices and edges. For each vertex \( v_i \in C \), we guess a time stamp \( T(v_i) \) in which \( v_i \) will be influenced. From Lemma 5, we know that \( T(v_i) \in \{0, 1, \ldots, 3t\} \).

There are \( t \) vertices in \( C \) and each one of them can be assigned any one of these \( 3t + 1 \) distinct values. So, there are \( (3t + 1)^t \) possible guesses for the time stamps of the vertices in \( C \). Now, among all the \( (3t + 1)^t \) possibilities, let us consider the
Let \((u, v)\) be an edge in \(G[B]\) such that \(\tau(u) = d_G(u)\) and \(\tau(v) = d_G(v)\). Without loss of generality assume that we have fixed the order \((u, v)\). Let \(F\) be the equivalence class such that \((u, v) \in F\). Note that for any \((u', v') \in F\), \(\tau(u) = d_G(u) = \tau(u') = d_G(u')\) and \(\tau(v) = d_G(v) = \tau(v') = d_G(v')\). Thus, by Observation 11, any target set contains at least one vertex from \(\{u, v\}\). So, for any such equivalence class \(F\), we have the following constraint.

\[
(x_{F,1} + x_{F,2} + x_{F,3}) = |F|
\]  

(2)
From the proof of Lemma 12, we have that for an equivalence class \( F \in \mathcal{Q} \) and \( u, v \in F \), \( g_j(u) = g_j(v) \) and \( h_j(u) = h_j(v) \). Therefore, we slightly abuse the notation and use \( g_j(F) \) and \( h_j(F) \) to denote \( g_j(u) \) and \( h_j(u) \), respectively, where \( u \in F \). Again from the proof of Lemma 12, we have that for an equivalence class \( F' \in \mathcal{P} \) and \((u_1, v_1), (u_2, v_2) \in F'\), \( g_j(u_1) = g_j(u_2), h_j(u_1) = h_j(u_2), g_j(v_1) = g_j(v_2), \) and \( h_j(v_1) = h_j(v_2) \). Therefore, we slightly abuse the notation and use \( g_j(F', 1), g_j(F', 2), h_j(F', 1), \) and \( h_j(F', 2) \) to denote \( g_j(u_1), g_j(v_1), h_j(u_1), h_j(v_1) \), respectively, where \((u_1, v_1) \in F'\).

For each vertex \( w \in C \) such that \( T_j(w) \geq 1 \), we construct a constraint. Towards that we need some notations. Let \( \mathcal{Q}_w \) be the set of equivalence classes in \( \mathcal{Q} \) such that for any vertex \( v \) in an equivalence class in \( \mathcal{Q}_w \), \( \{v, w\} \in E(G) \). Let \( \mathcal{P}_{w,1} \) be the set of equivalence classes in \( \mathcal{P} \) such that for any element \( (u, v) \) in an equivalence class in \( \mathcal{P}_{w,1}, \{u, w\} \in E(G) \), and \( \{v, w\} \notin E(G) \). Let \( \mathcal{P}_{w,2} \) be the set of equivalence classes in \( \mathcal{P} \) such that for any element \( (u, v) \) in an equivalence class in \( \mathcal{P}_{w,2}, \{u, w\} \notin E(G) \), and \( \{v, w\} \in E(G) \). Let \( \mathcal{P}_{w,3} \) be the set of equivalence classes in \( \mathcal{P} \) such that for any element \( (u, v) \) in an equivalence class in \( \mathcal{P}_{w,3}, \{u, w\}, \{v, w\} \in E(G) \). For a predicate \( Z \), \([Z]\) returns 1 if \( Z \) is true and 0 otherwise. The constraint we construct for \( w \) is the following.

\[
Y_w + Z_w + \sum_{v \in NG(w) \cap C} [T_j(v) < T_j(w)] \geq \tau(w) \tag{3}
\]

where \( Y_w \) and \( Z_w \) are defined as follows.

\[
Y_w = \sum_{F \in \mathcal{Q}_w} x_F + \sum_{F \in \mathcal{P}_{w,1}} (x_{F,1} + x_{F,3}) + \sum_{F \in \mathcal{P}_{w,2}} (x_{F,2} + x_{F,3})
+ \sum_{F \in \mathcal{P}_{w,3}} (x_{F,1} + x_{F,2} + 2x_{F,3})
\]

\[
Z_w = \sum_{F \in \mathcal{Q}_w} [g_j(F) < T_j(w)](|F| - x_F)
+ \sum_{F \in \mathcal{P}_{w,1}} [g_j(F, 1) < T_j(w)]x_{F,2} + [h_j(F, 1) < T_j(w)](|F| - x_{F,1} - x_{F,2} - x_{F,3})
+ \sum_{F \in \mathcal{P}_{w,2}} [g_j(F, 2) < T_j(w)]x_{F,1} + [h_j(F, 2) < T_j(w)](|F| - x_{F,1} - x_{F,2} - x_{F,3})
+ \sum_{F \in \mathcal{P}_{w,3}} \left( [g_j(F, 1) < T_j(w)]x_{F,2} + [g_j(F, 2) < T_j(w)]x_{F,1} + \right.
\left. ([h_j(F, 1) < T_j(w)] + [h_j(F, 2) < T_j(w)])(|F| - x_{F,1} - x_{F,2} - x_{F,3}) \right)
\]

Also, for any equivalence class \( F \in \mathcal{Q} \), we have

\[
0 \leq x_F \leq |F| \tag{4}
\]

For any equivalence class \( P \in \mathcal{P} \), we have

\[
(x_{P,1} + x_{P,2} + x_{P,3}) \leq |P| \tag{5}
\]

\[
x_{P,1}, x_{P,2}, x_{P,3} \geq 0 \tag{6}
\]
This completes the construction of the instance $I_j$. Recall that the number of equivalence classes is bounded by $2^{2t}(t + 1)^3$ and we have constructed at most 3 variables per equivalence class. This implies that the number of variables in $I_j$ is at most $3 \cdot 2^{2t}(t + 1)^3$. Hence, Condition (b) in the theorem is true. We have already mentioned that we constructed $(3t + 1)^t$ instances. Thus Condition (a) is satisfied.

In the rest of the proof, we show that Condition (c) is true. Let $S$ be a solution to the instance $(G, \tau, k, C)$ of TSS-D1M. Then there exits $j \in \{1, 2, \ldots, 3t\}$ such that $T_j : C \rightarrow \{0, 1, \ldots, 3t\}$ is a time stamp on $C$ by the target set $S$. Recall that $g_j$ is the output of Lemma 12 on the input $(G, \tau, C, T_j)$. Next we will show that $I_j$ is a YES-instance. Towards that we define values for variables of $I_j$ and prove this is a solution for $I_j$.

Let $F$ be an equivalence class in $\sim$. If the elements in $F$ are isolated vertices, then we set $x_F = |S \cap F|$. Suppose the elements in $F$ are isolated edges. Then, the values for $xF,1, xF,2$, and $xF,3$ are as follows. The value for $xF,1$ is the number of elements $(u, v)$ in $F$, where $u$ is included in $S$ and $v$ is not included in $S$. Similarly, the value for $xF,2$ is the number of elements $(u, v)$ in $F$, where $v$ is included in $S$ and $u$ is not included in $S$. The value for $xF,3$ is the number of elements $(u, v)$ in $F$, where both $u$ and $v$ are included in $S$. Since $|S| \leq k$, (1) is satisfied. By Observation 11 and the fact that $S$ is a target set, we have that (2) is satisfied. Let $F$ be an equivalence class in $\sim$ such that the elements in $F$ are singleton vertices. This implies that $N_G(x) = N_G(y)$ and $\tau(x) = \tau(y)$ for any $x, y \in F$. Thus, by the construction of $g_j$ in the proof of Lemma 12, we have that $g_j(x) = g_j(y) = g_j(F)$. Similarly, let $F'$ be an equivalence class in $\sim$ such that $(x_1, y_1), (x_2, y_2)$ be elements in $F'$. Then, again by the definition of equivalence classes in $\sim$ and the construction of $g_j$ and $h_j$, we have that $g_j(x_1) = g_j(x_2) = g_j(F', 1), g_j(y_1) = g_j(y_2) = g_j(F', 2), h_j(x_1) = h_j(x_2) = h_j(F', 1), and h_j(y_1) = h_j(y_2) = h_j(F', 2)$.

Now we prove that (3) is satisfied. Fix a vertex $w \in C$. Consider the L.H.S of (3). Notice that $Y_w$ is equal to the number of neighbors of $w$ in $B \cap S$. By Lemma 12, $Z_w$ is equal to the the number of neighbors of $w$ in $B \setminus S$ that are influenced strictly before the step $T_j(w)$. Thus, since $S$ is a target set and $w$ is influenced on step $T_j(w)$, (3) holds. It is easy to verify that (4)–(6) are satisfied. Thus, we conclude that $I_j$ is a YES-instance.

Now we prove the reverse direction of the proof. Suppose there is a YES-instance $I_j$. Then, we need to prove that $(G, \tau, k, C)$ is a YES-instance of TSS-D1M. As $I_j$ is a YES-instance, there are non-negative integer values for the variables $\{x_F : F \in Q\} \cup \{x_{P,1}, x_{P,2}, x_{P,3} : P \in P\}$ such that (1)–(6) are satisfied. From this we construct a set $S$ of size at most $k$ and prove that $S$ is a target set. Towards the construction of $S$, initially set $S := \emptyset$. Now consider an equivalence class $F \in Q$. We arbitrarily choose $x_F$ vertices from the equivalence class $F$ and add to $S$. Now consider an equivalence class $P \in P$. We choose arbitrary pairwise disjoint subsets $P_1, P_2, P_3$ of $P$ such that $|P_1| = x_{P,1}, |P_2| = x_{P,2},$ and $|P_3| = x_{P,3}$. For each element $(u, v) \in P_1$, we add $u$ to $S$. For each element $(u, v) \in P_2$, we add $v$ to $S$. For each element $(u, v) \in P_3$, we add both $u$ and $v$ to $S$. Finally we add $T_j^{-1}(0)$ to $S$. This completes the construction of $S$.

Next, we prove that $S$ is a target set of size at most $k$. The construction of $S$ and (1) implies that $|S| \leq k$. Because of (2) for any edge $(u, v) \in E(G[B])$ such that
\[ \tau(u) = d_G(u) \text{ and } \tau(v) = d_G(v), \text{ at least one of } u \text{ or } v \text{ is in } S. \] Because of this, from the proof of Lemma 12, for any \( u \in B \), \( h_j(u) \neq \infty \).

Now we define a function \( c : B \setminus S \mapsto \mathbb{N} \) as follows. For a vertex \( u \in B \setminus S \), if \( d_{G-C}(u) = 0 \), then we define \( c(u) = g_j(u) \). Consider a vertex \( u \in B \setminus S \) such that \( d_{G-C}(u) = 1 \). Let \( v \) be the only neighbor of \( u \) in \( G - C \). Then, if \( v \in S \), then we define \( c(u) = g_j(u) \) and otherwise \( c(u) = h_j(u) \). Since, for any \( u \in B \), \( h_j(u) \neq \infty \), we have that \( c(u) \neq \infty \).

Next using induction on \( i \), we prove that the vertices in the set \( T_{j-1}(i) \cup c^{-1}(i) \) are influenced on step \( i \) in the diffusion process starting from \( S \). The base case is when \( i = 1 \). Let \( w \) be a vertex in \( T_{j-1}(1) \). Consider (3). Note that \( Z_w = 0 \). Thus, we have that \( Y_w + \sum_{v \in N_G(w) \cap C} [T_j(v) < 1] \geq \tau(w) \). By the construction of \( S \), there are \( Y_w \) vertices in \( S \cap B \) that are neighbors of \( w \). Therefore, \( w \) is influenced on step 1. Let \( w' \in c^{-1}(1) \). Since \( T_{j-1}(0) \) are influenced on step 0 in the diffusion process starting at \( S \), by Lemma 12, \( w' \) is influenced on step 1 (because in a diffusion process, the vertices influenced on step 1 depends only on the seed set).

Now consider the induction step \( i > 1 \). Let \( S_{< i} \) be the set of vertices influenced before step \( i \). The construction of \( S \), (3) and the induction hypothesis imply that for any \( w \in T_{j-1}(i), |S_{< i} \cap N_G(w)| \geq \tau(w) \). Now, consider a vertex \( w' \in c^{-1}(i) \). By the induction hypothesis for any vertex \( x \in \bigcup_{r<i} T_{j-1}(r), x \) is influenced on step \( T_j(x) \). Thus, by Lemma 12 and the definition of the function \( c \), \( w' \) is influenced on step \( i \) (because in a diffusion process, the vertices influenced on step \( i \) depends only on the vertices influenced before step \( i \)).

Since \( c(u) \neq \infty \) for any \( u \in B \setminus C \), we have that all the vertices are influenced in the diffusion process starting from \( S \).

Theorem 13 and Proposition 10 imply Theorem 9.

### 5 Lower Bound when Parameterized by Treewidth

In this section, we show that for a graph on \( n \) vertices having treewidth \( k \) it is not possible to have an algorithm that solves the TSS problem in \( n^{o\left(\frac{k}{\log^2 k}\right)} \) time unless the Exponential Time Hypothesis (ETH) is false. Our proof is by a reduction from PARTITIONED SUBGRAPH ISOMORPHISM (PSI) stated below. The gadgets we use in our reduction and the arguments given are similar to the ones used by Ben-Zwi et al. in the proof of Theorem 1.2 in [16]. Given two graphs \( G \) and \( H \), an injective map \( \phi : V(G) \rightarrow V(H) \) is a subgraph isomorphism from \( G \) to \( H \), if for any \( \{u, v\} \in E(G) \), \( \{\phi(u), \phi(v)\} \in E(H) \).

**PARTITIONED SUBGRAPH ISOMORPHISM (PSI)**

**Input:** Two Graphs \( G \) and \( H \), a bijection \( f_G : V(G) \rightarrow \ell \), and a function \( c_H : V(H) \rightarrow \ell \), where \( |V(G)| = \ell \).

**Task:** Is there a subgraph isomorphism \( \phi \) from \( G \) to \( H \), such that for all \( v \in V(G) \), \( f_G(v) = c_H(\phi(v)) \)?
Theorem 14 (Corollary 6.3 [18]) If the PSI problem can be solved in \( f(G)n^{o\left(\frac{k}{\log k}\right)} \) time, where \( f \) is an arbitrary function, \( n = |V(H)| \), \( G \) is connected, and \( k \) is the number of edges of the smaller subgraph \( G \), then ETH fails.

Here, we present a reduction from the PSI problem to the TSS problem. We first describe a few notations that will be used in this section. In the PSI problem, we denote the larger graph by \( H \) and the smaller graph by \( G \), and let \( G' \) denote the graph for the TSS problem in the reduced instance. For an arbitrary color \( c \in \ell \), let \( V_c(H) \) denote the subset of vertices of \( H \) that are colored with the color \( c \). That is, \( V_c(H) = \{ v \in V(H) : c_H(v) = c \} \). Also, for any two distinct colors \( c_1, c_2 \in \ell \), we use \( E_{c_1, c_2}(H) \) to denote the subset of edges of \( H \) whose endpoints have colors \( c_1 \) and \( c_2 \). Now, we describe the construction (Fig. 1).

Construction In our construction, we use two types of gadgets, namely, selection gadgets and validation gadgets. The key idea of our construction is that any target set of \( G' \) encodes the vertices and edges of \( G \) in the partitioned subgraph isomorphism problem. Now, we describe the construction of the gadgets.

Selection Gadget: For each color class \( c \in \ell \), we construct a \( c \)-selection gadget which encodes the selection of a vertex of color \( c \) in \( H \). A \( c \)-selection gadget consists of a vertex \( x_v \) for every vertex \( v \in V_c(H) \). For a pair of distinct colors \( c_1, c_2 \in \ell \), if there exists an edge in \( G \) whose endpoints are colored \( c_1 \) and \( c_2 \), then we construct a \( (c_1, c_2) \)-selection gadget that encodes the selection of an edge in \( H \) whose endpoints are colored \( c_1 \) and \( c_2 \). A \( (c_1, c_2) \)-selection gadget consists of a vertex \( x_{\{u,v\}} \) for every edge \( \{u, v\} \in E_{c_1, c_2}(H) \). There are no edges among the vertices of the selection gadgets, and hence the union of these vertices forms an independent set in \( G' \). Finally, we add a guard vertex at each selection gadget.

![Fig. 1](image-url) An illustration of the validation gadget. Here, \( n = 5 \) and \( \text{low}(u) = 3 \).
that is connected to all the vertices in the gadget. Thus, each selection gadget is a star centered at the guard vertex.

**Validation Gadget:** We assign every vertex \( u \) of \( H \) two unique identifiers denoted as \( \text{high}(u) \) and \( \text{low}(u) \) with \( \text{low}(u) \in [n] \) and \( \text{high}(u) = 2n - \text{low}(u) \), where \( n = |V(H)| \). Let \( c_1, c_2 \in [\ell] \) be two distinct colors for which a \((c_1, c_2)\)-selection gadget is present. We construct a validation gadget between (i) the \((c_1, c_2)\)-selection gadget and the \(c_1\)-selection gadget, and (ii) the \((c_1, c_2)\)-selection gadget and the \(c_2\)-selection gadget. We describe the validation gadget between \(c_1\) - and \((c_1, c_2)\)-selection gadgets. It consists of two vertices, the validation-pair. The first vertex of this pair has \( \text{high}(u) \) number of internally vertex disjoint 2-length paths to each vertex \( x_u, u \in V_{c_1}(H) \). Each such 2-length path has exactly one internal vertex which is called a connection vertex. Similarly, the first vertex in the validation pair has \( \text{low}(u) \) number of internally vertex disjoint 2-length paths (whose internal vertices are called connection vertices as before) to edge selection gadget vertex \( x_{[u,v]}, \{u, v\} \in E_{c_1,c_2}(H) \) and \( u \in V_{c_1}(H) \). Now we come to the edges emanating from the second vertex in the validation pair. The second vertex of this pair has \( \text{low}(u) \) number of internally vertex disjoint 2-length paths to each vertex \( x_u, u \in V_{c_1}(H) \). As before, the internal vertices in such path are called connection vertices. Similarly, the second vertex in the validation pair has \( \text{high}(u) \) number of internally vertex disjoint 2-length paths to edge selection gadget vertex \( x_{[u,v]}, \{u, v\} \in E_{c_1,c_2}(H) \) and \( u \in V_{c_1}(H) \).

See Fig. 1 for an illustration of validation and selection gadgets. Let \( |V(G)| = \ell \) and \( |E(G)| = k \). From our construction, we have \( \ell \) vertex selection gadgets corresponding to each color class \( c \in [\ell] \) and \( k \) edge selection gadgets corresponding to each edge in \( G \). There are \( 2k \) validation gadgets. We complete the construction by assigning a diffusion threshold to all the vertices in \( G' \). All the guard vertices and the connection vertices will have a threshold of 1. All the vertex and edge selection vertices will have thresholds equal to their degrees in \( G' \). Finally, the vertices in the validation pairs will have a threshold of \( 2n \).

**Lemma 15** An instance \((G, H, f_G, c_H)\) of the PSI problem, where \( |V(G)| = \ell \), \( |E(G)| = k \), and \( |V(H)| = n \), is a YES instance if and only if \( G' \) has a target set of size \( \ell + k \).

**Proof** First, we prove the forward direction. Assume that we are given a YES-instance of the PSI problem. Let

\[
S = \{v \in V(H) : v \text{ is the image of some vertex in } G \text{ under } \phi \},
\]

\[
S' = \{x_v : v \in S\} \cup \{x_{[u,v]} : u, v \in S, \{\phi^{-1}(u), \phi^{-1}(v)\} \in E(G)\}.
\]

Observe that \( |S'| = \ell + k \). We claim that \( S' \) is a target set for \( G' \). We prove this claim by explaining the diffusion process. The vertices in the seed set \( S' \) are influenced at time \( t = 0 \). At time \( t = 1 \), all the guard vertices along with connection vertices that are adjacent to the vertices in \( S' \) will be influenced as their thresholds are 1. At time \( t = 2 \), the vertices of every validation pair will be influenced. By time \( t = 3 \), the remaining connection vertices will be influenced. Since all the guard
vertices were influenced at \( t = 1 \), this would mean that all the remaining vertex and edge selection gadget vertices will be influenced by time \( t = 4 \). This completes the proof of the forward direction.

Now, we prove the reverse direction. Assume that \( G' \) contains a target set \( S' \) of size \( \ell + k \). We need to show that in the PSI instance, there exists a subgraph isomorphism \( \phi \) from \( G \) to \( H \) such that for all \( v \in V(G) \), \( f_G(v) = c_H(\phi(v)) \). We can assume, without loss of generality, that the target set \( S' \) does not contain any of the guard vertices. This is because, in place of the guard vertices if we select appropriate selection vertices, we can still influence all the vertices of \( G' \). As the guard vertices are only connected with the vertex and edge selection gadget vertices, before the guard vertices are influenced, there has to be at least one active vertex in each selection gadget. Further, no vertex in a selection gadget can be influenced if the guard vertex adjacent to it is not active. This is because the vertices inside the selection gadgets have their thresholds equal to their degrees. We thus have a deadlock situation where an inactive guard vertex of a selection gadget is looking for an active guard vertex. The only way to resolve this is to have at least one vertex from each selection gadget in \( S' \). Since it is given that the size of the target set \( S' \) is equal to \( \ell + k \), we have exactly one vertex from each selection gadget in \( S' \). Further, for any distinct \( u, v \in V(H) \), since either \( \text{low}(u) + \text{high}(v) \) or \( \text{low}(v) + \text{high}(u) \) is less than \( 2n \), the only way to activate a validation pair is to choose a pair of vertices (one each from the vertex selection gadget and the edge selection gadget in its neighborhood) corresponding to an incident vertex and edge pair in \( H \). Thus, the endpoints of the edges of \( H \) selected in the edge-selection gadgets of \( G' \) are vertices of \( H \) selected in the vertex selection gadgets of \( G' \). So corresponding to the set \( S' \) we get a unique subgraph \( H_{S'} \) of \( H \) with \( V(H_{S'}) = \{ v \in V(H) : x_v \in S' \} \) and \( E(H_{S'}) = \{ (u, v) \in E(H) : x_{[u,v]} \in S' \} \). Observe that \( H_{S'} \) has exactly \( \ell \) vertices and \( k \) edges. Further, no two vertices of \( H_{S'} \) are of the same color as \( S' \) contains exactly one vertex from each selection gadget. That is, for any two distinct \( u, v \in V(H_{S'}) \), \( c_H(u) \neq c_H(v) \). Let \( c \in [\ell] \). We define a map \( \phi : V(G) \to V(H_{S'}) \) as: for a vertex \( u \in V(G) \) with \( f_G(u) = c \), we have \( \phi(u) = v \) if \( x_v \) is that vertex in \( S' \) that belongs to the \( c \)-selection gadget in \( G' \). Clearly, \( c_H(v) = c = f_G(u) \). Since no two vertices of \( G \) or \( H_{S'} \) are of the same color and every vertex of \( G \) is mapped to a vertex of its own color in \( H_{S'} \) under \( \phi \), the map \( \phi \) is injective. Further, since \( |V(H_{S'})| = |V(G)| = \ell \), \( \phi \) is a bijective map. What is left is to show that \( \phi \) is a subgraph isomorphism from \( G \) to \( H \). Suppose \( {u_1, u_2} \in E(G) \) with \( f_G(u_1) = c_1 \) and \( f_G(u_2) = c_2 \). Let \( v_1 := \phi(u_1) \) and \( v_2 := \phi(u_2) \). Then, from what we have already shown, \( c_H(v_1) = c_1 \) and \( c_H(v_2) = c_2 \). Further, \( x_{v_1} \) (or \( x_{v_2} \)) is the only vertices from the \( c_1 \)-selection (or \( c_2 \)-selection) gadget that was selected in \( S' \). Since there is an edge in \( G \) whose endpoints are colored with \( c_1 \) and \( c_2 \), \( G' \) has a \((c_1, c_2)\)-selection gadget. For the validation pair between \( c_1 \)-selection gadget and \((c_1, c_2)\)-selection gadget and the one between \( c_2 \)-selection gadget and \((c_1, c_2)\)-selection gadget to be activated, the one vertex which is selected in \( S' \) from the \((c_1, c_2)\)-selection gadget has to be \( x_{v_1, v_2} \). This implies that \( \{v_1, v_2\} \in E(H_{S'}) \). Thus, \( \phi \) is a subgraph isomorphism and the given instance is a YES instance of the PSI problem.
The minimum number of vertices of a graph $G$ whose removal results in a graph that is a disjoint collection of star graphs is called the \textit{star-deletion number} of $G$.

\textbf{Lemma 16} \ The graph $G'$ has a star-deletion set of size $O(k)$.

\textit{Proof} \ Observe that removing all the guard vertices and all the validation pairs from $G'$ results in a disjoint collection of star graph. Thus $G'$ has a star-deletion number of size at most $\ell + k + 4k = O(k)$, as $G$ is connected implies $\ell \leq k + 1$. \hfill $\square$

Below, we have the main theorem of this section which follows from Theorem 14 and Lemmas 15 and 16.

\textbf{Theorem 17} \ For a graph on $n$ vertices having a star-deletion number of size $k$, it is impossible to have an algorithm that solves the TSS problem in $n^{o(\frac{1}{\log k})}$ time unless ETH is false.

\textit{Proof} \ Suppose there is an algorithm $A$ that solves the TSS problem on a graph on $s$ vertices having a star-deletion set of size $p$ in $s^{o(\frac{p}{\log p})}$ time. Consider an instance $(G, H, f_G, c_H)$ of the PSI problem, where $|V(G)| = \ell$, $|E(G)| = k$, $G$ is connected, and $|V(H)| = n$. If $\ell > n$, then clearly it is a NO instance. So assume $\ell \leq n$. From this instance of the PSI Problem, construct an instance $(G', k')$ of the TSS problem as outlined in the beginning of this section, where $N := |V(G')| = O(n^3)$ and $k' = \ell + k$. Such a construction can be done in polynomial time. Lemma 15 states that $(G, H, f_G, c_H)$ is a YES instance of the PSI problem if and only if $(G', k')$ is a YES instance of the TSS Problem. Further, from Lemma 16, we know that $G'$ has a star-deletion set of size $O(k)$. We use algorithm $A$ to solve the TSS problem instance $(G', k')$ in $N^{o(\frac{k}{\log k})}$ time. But, this solves the PSI problem instance $(G, H, f_G, c_H)$ in $n^{o(\frac{\ell}{\log \ell})}$ time which by Theorem 14 is not possible unless ETH is false. \hfill $\square$

The following theorem is due to Ben-Zwi et al.

\textbf{Theorem 18} (Theorem 1.2, Ben-Zwi et al. [16]) \ For a graph on $n$ vertices having a treewidth of $w$, it is impossible to have an algorithm that solves the TSS problem in $n^{o(\sqrt{w})}$ time unless all problems in SNP can be solved in sub-exponential time.

Since the tree-depth of a graph is at most its star-deletion number plus one and the treewidth is at most tree-depth, we have the following corollary to Theorem 17 which is an improvement of the lower bound to the complexity of the TSS problem given in Theorem 18.

\textbf{Corollary 19} \ For a graph on $n$ vertices having a tree-depth of $w$ (and hence treewidth is at most $w$), it is impossible to have an algorithm that solves the TSS problem in $n^{o(\frac{w}{\log w})}$ time unless ETH is false.
6 Conclusion

In this work, we improve the running time of TSS-D0M to $2^{O(t \log t)} n$ by reducing the problem to MULTI-HITTING SET and solving the latter. We believe that the MULTI-HITTING SET problem could be of independent interest. We have proved that TSS-D1M is FPT. An open question we propose here is the parameterized complexity of TSS-D2M.

References

1. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: Proceedings of the 21st International Conference on World Wide Web, pp. 519–528. ACM (2012)
2. Hu, Y.-C., Perrig, A., Johnson, D.B.: Wormhole attacks in wireless networks. IEEE J. Selected Areas Commun. 24(2), 370–380 (2006)
3. Salathé, M., Kazandjieva, M., Lee, J.W., Levis, P., Feldman, M.W., Jones, J.H.: A high-resolution human contact network for infectious disease transmission. Proceedings of the National Academy of Sciences, 201009094 (2010)
4. Chen, N.: On the approximability of influence in social networks. SIAM J. Discret. Math. 23(3), 1400–1415 (2009)
5. Chiang, C.-Y., Huang, L.-H., Li, B.-J., Wu, J., Yeh, H.-G.: Some results on the target set selection problem. J. Comb. Optim. 28(4), 702–715 (2013)
6. Cicalese, F., Cordasco, G., Gargano, L., Milanič, M., Vaccaro, U.: Latency-bounded target set selection in social networks. Theor. Comput. Sci. 535, 1–15 (2014)
7. Chopin, M., Nichterlein, A., Niedermeier, R., Weller, M.: Constant thresholds can make target set selection tractable. Theory Comput. Syst. 55(1), 61–83 (2014)
8. Dvorák, P., Knop, D., Toufar, T.: Target set selection in dense graph classes. In: Hsu, W., Lee, D., Liao, C. (eds.) 29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan. LIPIcs, vol. 123, pp. 18–11813. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018). https://doi.org/10.4230/LIPIcs.ISAAC.2018.18
9. Bazgan, C., Chopin, M., Nichterlein, A., Sikora, F.: Parameterized inapproximability of target set selection and generalizations. Computability 3(2), 135–145 (2014)
10. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. Soc. Netw. Anal. Min. 3(2), 233–256 (2013)
11. Hartmann, T.A.: Target set selection parameterized by clique-width and maximum threshold. In: International Conference on Current Trends in Theory and Practice of Informatics, pp. 137–149. Springer (2018)
12. Bliznets, I., Sagunov, D.: Solving target set selection with bounded thresholds faster than $2^n$. In: Paul, C., Pål, M., (eds.) 13th International Symposium on Parameterized and Exact Computation (IPEC 2018), Leibniz International Proceedings in Informatics (LIPIcs), vol. 115, pp. 22–12214. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.IPEC.2018.22
13. Keiler, L., Lima, C.V.G., Maia, A.K., Sampaio, R., Sau, I.: Target set selection with maximum activation time. arXiv:2007.05246 (2020)
14. Knop, D., Schierreich, S., Suchý, O.: Balancing the spread of two opinions in sparse social networks. arXiv:2105.10184 (2021)
15. Banerjee, S., Mathew, R., Panolan, F.: Target set selection parameterized by vertex cover and more. arXiv:1812.01482 (2021)
16. Ben-Zwi, O., Hermelin, D., Lokshonov, D., Newman, I.: Treewidth governs the complexity of target set selection. Discret. Optim. 8(1), 87–96 (2011)
17. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshonov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms, vol. 4 Springer (2015)
18. Marx, D.: Can you beat treewidth? In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07), pp. 169–179. IEEE (2007)
19. Lenstra Jr., H.: Integer programming with a fixed number of variables. Math. Oper. Res. 8(4), 538–548 (1983). https://doi.org/10.1287/moor.8.4.538
20. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing. STOC ’83, pp. 193–206. Association for Computing Machinery (1983). https://doi.org/10.1145/80061.808749

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Affiliations

Suman Banerjee1 · Rogers Mathew2 · Fahad Panolan2

Suman Banerjee
suman.banerjee@iitjammu.ac.in

Rogers Mathew
rogers@cse.iith.ac.in

1 Department of Computer Science and Engineering, Indian Institute of Technology Jammu, Jammu & Kashmir, 181221, India
2 Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Sangareddy, 502284, Telangana, India