Target Set Selection on Graphs of Bounded Vertex Cover Number

Suman Banerjee[*1], Rogers Mathew[2], and Fahad Panolan[2]

[1] Department of Computer Science and Engineering,
Indian Institute of Technology, Gandhinagar, India.
`suman.b@iitgn.ac.in`

[2] Department of Computer Science and Engineering,
Indian Institute of Technology, Hydrabad, India.
`{rogers, fahad }@cse.iith.ac.in`

Abstract. Given a simple, undirected graph G with a threshold function $\tau : V(G) \to \mathbb{N}$, 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. We have the following results on the TSS Problem:

– It was shown by Nichterlein et al. [Social Network Analysis and Mining, 2013] that it is possible to compute an optimal sized target set in $O(2^{(2^t+1)t} \cdot m)$ time, where m and t denote the number of edges and the cardinality of a minimum vertex cover, respectively, of the graph under consideration. We improve this result by designing an algorithm that computes an optimal sized target set in $2^{O(t \log t)} n^{O(1)}$ time, where n denotes the number of vertices of the graph under consideration.

– We show that the TSS Problem on bipartite graphs does not admit an approximation algorithm with a performance guarantee asymptotically better than $O(\log n_{min})$, where n_{min} is the cardinality of the smaller bipartition, unless $P = NP$. Chen et al. [SIDMA, 2009] had shown that the TSS Problem on general graphs does not admit an approximation algorithm with a performance guarantee asymptotically better than $O(2^{\log^{1-\epsilon} n})$, where n is the number of vertices of the graph under consideration, unless $NP \subseteq DTIME(n^{polylog(n)})$.

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 [1,7]; propagation of virus, wormhole in a computer network [10]; spreading of contaminating diseases in a human contact network [12], 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 some non-negative 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.

1.1   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., $\mathcal{A}[S,0] \subseteq \mathcal{A}[S,1] \subseteq \mathcal{A}[S,2] \subseteq \cdots \subseteq \mathcal{A}[S,i] \subseteq \cdots \subseteq V(G)$, where $\mathcal{A}[S,i]$ denotes the set of nodes that has been influenced on or before the ith time stamp and $\mathcal{A}[S,0] = S$. For all $i > 0$, the diffusion process can be expressed by the following equation:

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

where $N(u)$ denotes the set of neighbors of u. For any seed set S, we define $\mathsf{influence}_G(S) := \bigcup_{t \geq 0} \mathcal{A}[S,t]$. Observe that $t \leq |V(G)|$ as at least one new node is activated in every time step; else the diffusion process stops. The TARGET SET SELECTION Problem is about finding a minimum cardinality target set. In other words, it is about finding a minimum cardinality seed set S such that $\mathsf{influence}_G(S) = V(G)$.

> **TSS Problem (Optimization Version)**
>
> **Instance:** *An undirected graph G with a threshold function*
> $$\tau : V(G) \to \mathbb{N}.$$
>
> **Problem:** *Find a minimum cardinality target set $S \subseteq V(G)$ such that* $\mathsf{influence}_G(S) = V(G)$.

1.2   Related work

Chen [4] showed that the TSS Problem cannot be approximated within a factor of $O(2^{\log^{1-\epsilon} n})$ of the optimum for a fixed constant $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$, by a reduction from the MINREP Problem. They also showed that the TSS Problem 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. In [5], Chiang et al. showed that the TSS Problem can be solved in linear time for block-cactus graphs with an arbitrary threshold and for chordal graph with threshold at most 2. In [6], Chiang et al. studied the TSS Problem on *Honeycomb Networks* under the *majority threshold*, where the threshold value of each node is more than half of its degree. They gave the exact value of the size of an optimal target set for different types of honeycomb networks under a strict majority threshold model. Chopin et al. [8] 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 the TSS Problem 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. [9] added a few more results in the parameterized setting. They showed that the TSS Problem is W[1]-hard with respect to parameter *neighborhood diversity*, and under majority threshold this problem has an FPT algorithm with respect to the parameters *neighborhood diversity*, *twin cover*, *modular width*. Bazgan et al. [2] showed that for any function $f(\cdot)$ and $\rho(\cdot)$ this problem cannot be approximated within a factor of $\rho(k)$ in $f(k) \cdot n^{O(1)}$ time unless $FPT = W[P]$ even for constant and majority thresholds, where k denotes the cardinality of the target set. Nichterlein et al. [11] showed that the TSS Problem can be approximated in polynomial time on trees and bounded-degree graphs. They showed that for diameter two split graphs the TSS Problem remains W[2]-hard with respect to the parameter *size of the target set*. Also, TSS Problem is fixed parameter tractable when parameterized by the *vertex cover number* and *cluster editing number*. Bliznets et al. [3] 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.

1.3   Our contribution

Let G be a graph on n vertices and let t be the size of a minimum vertex cover of G. Let $\tau : V(G) \to \mathbb{N}$ denote the threshold function. In [11], Nichterlein et al. showed that the TSS Problem admits an FPT algorithm parameterized by the size of a minimum vertex cover. More specifically, they showed that the TSS Problem on G can be solved in $\mathcal{O}(2^{(2^t+1)t}.m)$ time, where m denotes the number of edges of G. In Theorem 3 in Section 2, we improve this result to show that the TSS Problem on G can be solved in $2^{O(t \log t)} n^{O(1)}$ time.

In Theorem 5 in Section 3, we show that unless $P = NP$, if the underlying influence graph is bipartite, the optimization version of the TSS Problem cannot have a polynomial time approximation algorithm with a performance guarantee better than $O(\log n_{min})$, where n_{min} is the cardinality of the smaller part in the bipartition. Chen et al. [4] had shown that, for any $\epsilon > 0$, the TSS Problem on general graphs cannot have a polynomial time approximation algorithm with a performance guarantee better than $O(2^{\log^{1-\epsilon} n})$, where n is the number of vertices of the graph under consideration, under a stronger assumption that $NP \not\subset QP$, where, $QP = DTIME(n^{polylog(n)})$, stands for the complexity class *Quasi Polynomial*. The reader may note that it is possible to extend Chen et al.'s inapproximability result on general graphs to bipartite graphs using a self reduction described below: subdivide every edge exactly once (in other words, replace every edge with a path of length 2) in the given general graph on n vertices and m edges to create a new graph having $n+m$ vertices and $2m$ edges. The new graph is bipartite as it does not contain any odd cycle. Retain the threshold values of the old vertices; for each of the m new vertices, assign a threshold value of 1. This gives a $\Omega(2^{\log^{1-\epsilon} \sqrt{n}})$ inapproximability result for the TSS Problem on bipartite graphs, under the stronger assumption that $NP \not\subset QP$.

1.4   Preliminaries

Throughout the paper, we consider finite, undirected and simple graphs. For any vertex v in the graph under consideration (which should be clear from the context), we shall use $deg(v)$ to denote the number of edges incident on v and $N(v)$ to denote the set of vertices adjacent to v. 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. A set $S \subseteq V(G)$ is said to be a *vertex cover* of G, if for every edge in G at least one of its end-vertices is present in S. The size of a minimum vertex cover of G is called the vertex cover number of G. A set $T \subseteq V(G)$ is said to be an *independent set* of G if no two vertices in T are adjacent with each other.

2 FPT algorithm parameterized by vertex cover number

Consider the target set selection problem on a graph G with a threshold function $\tau : V(G) \to \mathbb{N}$. Clearly, all the vertices v having $\tau(v) > deg(v)$ are present in every feasible target set. So while designing an algorithm to find an optimal target set in G, it is safe to include all such vertices into the solution set to be constructed. Throughout this section we therefore assume that the graph G under consideration has $\tau(v) \leq deg(v)$, for all $v \in V(G)$. We begin by stating the following easy-to-see remark.

Remark 1. Let C be a vertex cover of a graph G with a threshold function $\tau : V(G) \to \mathbb{N}$. Then, C is a target set in G.

Lemma 1. *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 $C \cap \mathsf{influence}_G(S)$ will be influenced in at most $2t - 1$ steps. Therefore, the diffusion process will terminate in at most $2t$ steps.

We now show a Turing reduction from the decision version of the target set selection Problem to *Multi-Hitting Set Problem*. These two problems are defined as follows.

TSS (Decision Version)   **Parameter:** t
Input: An undirected graph G on n vertices, a threshold function $\tau : V(G) \mapsto \mathbb{N}$, $k \in \mathbb{N}$, and a vertex cover C of size t.
Question: Is there a target set of size k?

Multi-Hitting Set   **Parameter:** t
Input: A universe U, where $|U| \leq n$, a collection of subsets $S_1, S_2, \ldots, S_t \subseteq U$ and $q, l_1, l_2, \ldots, l_t \in \mathbb{N}$ such that $l_j \leq t$, for all $j \in [t]$.
Question: Is there a subset $H \subseteq U$, such that $|H| \leq q$ and $|H \cap S_i| \geq l_i$, for all $i \in [t]$.

Now, we describe our Turing reduction that constructs $2^{O(t \log t)}$ instances of Multi-Hitting Set from a given instance of TSS (Decision Version).

Theorem 1. *There is an algorithm that given an instance (G, τ, k, C) of TSS (Decision Version), where $t = |C|$, runs in $2^{O(t \log t)} \cdot n^{O(1)}$ time, and outputs*

a collection of instances $\mathcal{I} = \{I^j = (V(G) \setminus C, q^j, S_1^j, S_2^j, \ldots, S_t^j, l_1^j, l_2^j, \ldots, l_t^j) : j \in [s]\}$ *of* MULTI-HITTING SET *such that the following holds:*

- *(a) The number of instances, i.e., $s \in 2^{\mathcal{O}(t \log t)}$.*
- *(b) (G, τ, k, C) is a YES-instance of* TSS (DECISION VERSION) *if and only if there exists $j \in [s]$ such that I^j is a YES-instance of* MULTI-HITTING SET.*

Proof. First we describe the construction of $2^{\mathcal{O}(t \log t)}$ many instances of MULTI-HITTING SET from a given instance (G, τ, k, C) of TSS (DECISION VERSION).

Construction. Consider the given TSS (DECISION VERSION) instance (G, τ, k, C), where G is a graph on n vertices and $C := \{v_1, \ldots, v_t\}$ is a vertex cover of G of size t. Let $B = V(G) \setminus C$. We know B is an independent set in G. For each vertex v_i in C, we guess a time stamp $T(v_i)$ in which v_i will be influenced. From Lemma 1, we know that $T(v_i) \in \{0, 1, \ldots, 2t\}$. 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 $t^{2t+1} = 2^{\mathcal{O}(t \log t)}$ possible guesses for the time stamps of the vertices in C. Now, among all the $2^{\mathcal{O}(t \log t)}$ 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 vertex $u \in B$, 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 vertex $v_i \in C$, we define $l_i^j := \max\{0, \tau(v_i) - |\{w \in N(v_i) : T^j(w) < T^j(v_i)\}|\}$ and $S_i^j := \{w \in N_B(v_i) : T^j(w) \geq T^j(v_i)\}$. We remark that if $l_i^j \geq t$, 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_i^j's and l_i^j'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 $2^{O(t \log t)}$. The following claims are used to prove the correctness of the reduction (i.e., to prove property (b)).

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 $i \in [t], |S \cap S_i^j| \geq l_i^j$. 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 l_i^j vertices from S_i^j should be there in the target set S. This implies that $S \cap B$ is a solution for the instance I^j.

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 q that all the vertices $v_i \in C$ with $T^j(v_i) \leq q$ will be influenced

by the end of step q. The base case is when $q \leq 1$. Clearly, for $q = 0$, all the vertices $v_i \in C$ with $T^j(v_i) \leq q$ 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 $l_i^j = \tau(v_i)$ 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 $q > 1$. Now consider a vertex $v_i \in C$ with $T^j(v_i) = q$. We know that at least l_i^j vertices from $S_i^j := \{w \in N_B(v_i) \; : \; T^j(w) \geq q\}$ are present in H because H is a solution to I^j. By the induction hypothesis, we have that for any vertex $v_r \in C$ with $T^j(v_r) = q - 2$ is influenced at the end of step $q - 2$. This implies that all the vertices in $\{w \in N_B(v_i) \; : \; T^j(w) < q\}$ are influenced by the end of step $q - 1$. Therefore, at least $\tau(v_i)$ vertices from $N(u)$ will be influence by the end of step $q - 1$. This implies that v_i will be influenced in step q. This completes the proof of the claim.

Now we are ready to prove the correctness of the algorithm. Assume that (G, τ, k, C) is a YES-instance, which means that G has a target set S of size k. Recall that $C = \{v_1, \ldots, v_t\}$ is a vertex cover of size t in G and let $B = V(G) \setminus C$. Let S_C and S_B denote the vertices of S from the part C and B, respectively, i.e., $S = S_C \uplus S_B$. Let $T^S(v_i)$ be the time stamp at which a vertex $v_i \in C$ is activated when the diffusion process is initiated with S as the seed set. Consider the construction of that j-th instance $I^j = (V(\mathcal{G}) \setminus C, q^j, S_1^j, S_2^j, \ldots, S_t^j; l_1^j, l_2^j, \ldots, l_t^j)$ of the multi-hitting set problem where $T^j(v_i) = T^S(v_i)$, for all $v_i \in C$. Then by Claim 1 S_B is a hitting set for the instance I^j.

Now, consider the reverse direction of the proof. Suppose there is a $j \in [s]$ such that I^j is a YES-instance of MULTI-HITTING SET. That is, H^j is a hitting set for I^j with $|H^j| \leq q^j = k - \{v_i \in C \; : \; T^j(v_i) = 0\}$. Then, from Claim 2, $H^j \cup \{v_i \in C \; : \; T^j(v_i) = 0\}$ is a target set for G of cardinality at most k.

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

Theorem 2. *Let $I = (U, q, S_1, S_2, \ldots, S_t; l_1, l_2, \ldots, l_t))$ be an instance of* MULTI-HITTING SET. *Then, I can be decided in $2^{O(t \log t)} \cdot n^{O(1)}$ time, 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 $D_j(q', l_1', l_2', \ldots, l_t')$ 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 l_i' elements, where $0 \leq l_i' \leq l_i$, for all $i \in [t]$ and $0 \leq q' \leq q$. The case when $j = 0$ can be computed easily as follows.

$$D_0(q', l_1', l_2', \ldots, l_t') = \begin{cases} \emptyset \text{ if } l_i' = 0 \text{ for all } i, \\ NULL \text{ otherwise.} \end{cases}$$

We compute the DP table entries in the 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 $q', l_1', l_2', \ldots, l_t'$ such that $0 \leq l_i' \leq l_i$,

for all $i \in [t]$ and $0 \le q' \le q$, we compute $D_j(q', l'_1, l'_2, \ldots, l'_t)$ as follows. If $D_{j-1}(q', l'_1, l'_2, \ldots, l'_t) \ne NULL$, then

$$D_j(q', l'_1, l'_2, \ldots, l'_t) = D_{j-1}(q', l'_1, l'_2, \ldots, l'_t).$$

If $D_{j-1}(q' - 1, l'_1 - 1, l'_2 - 1, \ldots, l'_k - 1, l'_{k+1}, \ldots, l'_t) \ne NULL$, then set

$$D_j(q', l'_1, l'_2, \ldots, l'_t) = D_{j-1}(q' - 1, l'_1 - 1, l'_2 - 1, \ldots, l'_k - 1, l'_{k+1}, \ldots, l'_t) \cup \{j + 1\}.$$

Otherwise, we set $D_{j+1}(q', l'_1, l'_2, \ldots, l'_t) = $ NULL. Using this dynamic programming approach, we eventually compute $D_n(q, l_1, \ldots, l_t)$. Since $w \le n$ and each of $q, l_1, \ldots, l_t$ is at most t, we can compute this in time $(t + 1)^t n^{O(1)} = 2^{O(t \log t)} \cdot n^{O(1)}$.

Next we prove the correctness of our algorithm. Towards that we claim that $D_n(q, l_1, \ldots, l_t) = $ NULL if and only if I is a NO-instance. We prove this by proving a more general statement. We will show that, for every $j \in \{0, \ldots, n\}$, $D_j(q', l'_1, \ldots, l'_t) = $ NULL if and only if $I_j(l'_1, \ldots, l'_t) :=$ $(U_j, q', S'_1, \ldots, S'_t; l'_1, \ldots, l'_t)$ is a NO-instance where $S'_i = S_i \cap U_j$, $l'_i \le l_i$, for every i, and $0 \le q' \le q$. 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', l'_1, \ldots, l'_t) = $ NULL. This implies, both $D_{j-1}(q' - 1, l'_1 - 1, \ldots, l'_k - 1, l'_{k+1}, \ldots, l'_t) = $ NULL and $D_{j-1}(q', l'_1, \ldots, l'_t) = $ NULL. Thus, by induction hypothesis, there is no multi-hitting set of size q' for the instance $I_{j-1}(l'_1, \ldots, l'_t)$ and there is no multi-hitting set of size $q' - 1$ for the instance $I_{j-1}(l'_1 - 1, \ldots, l'_k, \ldots, l'_t)$. This implies that there is no multi-hitting set of size q' that contains element u_j and there is no feasible multi-hitting set of size q' for instance I' that does not contain element u_j for the instance $I_j(l'_1, \ldots, l'_t)$. Hence, $I_j(l'_1, \ldots, l'_t)$ is a NO-instance. Now, to prove the reverse direction of the bidirectional statement, assume that $I_j(l'_1, \ldots, l'_t)$ is a NO-instance. Then, clearly $(U_{j-1}, q' - 1, S'_1 \setminus u_j, \ldots, S'_k \setminus \{u_j\}, S'_{k+1}, \ldots, S'_t; l'_1 - 1, \ldots l'_k - 1, l'_{k+1}, \ldots, l'_t)$ and $(U_{j-1}, q', S'_1, \ldots, S'_t; l'_1, \ldots, l'_t)$ are NO-instances and therefore both $D_{j-1}(q' - 1, l'_1 - 1, \ldots, l'_k - 1, l'_{k+1}, \ldots, l'_t) = NULL$ and $D_{j-1}(q', l'_1, \ldots, l'_t) = $ NULL. Hence, $D_j(q', l'_1, \ldots, l'_t) = $ NULL. This proves the theorem.

Below we state the main result of this section which follows directly from Theorems 1 and 2 and from the fact that a polynomial time 2-factor approximation algorithm exists for computing a minimum vertex cover in a graph.

Theorem 3. *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 comver in G. Then, the optimal target set for G can be computed in time $2^{O(t \log t)} n^{O(1)}$.*

3 Inapproximability result for the TSS Problem on bipartite graphs

We state and prove an auxiliary lemma below which is required to establish the inapproxibility result in Section 3.1.

Lemma 2. *Let G be a bipartite graph with bipartition $\{V_1, V_2\}$. Let $\tau : V(G) \to \mathbb{N}$ be defined as $\tau(v) = 1$, if $v \in V_1$ and $\tau(v) = deg(v)$, if $v \in V_2$. Further, it is given that the degree of every vertex in G is at least 1. Then given any feasible solution S for the TSS Problem, we can generate another feasible solution S' in polynomial time with $|S'| \leq |S|$ that satisfies the following two properties: (i) $S' \subseteq V_2$ and (ii) $\forall v \in V_1, N(v) \cap S' \neq \emptyset$.*

Proof. If the given feasible solution S is a subset of V_2, then $S' = S$. Otherwise, we construct S' from S by following the iterative procedure given below. Initialize $S' = S \cap V_2$. We pick a vertex u from $S \cap V_1$. If $N(u) \cap S' = \emptyset$, then pick a vertex from $N(u)$ (say, $v \in N(u)$) and update S' as $S' = S' \cup \{v\}$. If we perform this operation for all the nodes of $S \cap V_1$, we get a set S' with $S' \subseteq V_2$. It is easy to observe that this operation can be performed in polynomial time. Since for each node $u \in S \cap V_1$, we choose at most one vertex from V_2 to include it in the set S', we have, $S' \subseteq V_2$ and $|S'| \leq |S|$. Now, we argue that S' is also a feasible solution.

Using the fact that S is a feasible solution for the TSS Problem on G, we have the following observation:

– **Observation:** For all $u \in V_1 \setminus S$, $N(u) \cap (V_2 \cap S) \neq \emptyset$.

From this observation, we can say that $\forall u \in V_1 \setminus S$, $N(u) \cap S' \neq \emptyset$. Also $\forall u \in V_1 \cap S$, at least one of its neighbors is in S'. Hence, we can say $\forall u \in V_1$, $N(u) \cap S' \neq \emptyset$. As, $\forall u \in V_1$, $\tau(u) = 1$. The nodes in S' will be able to influence all the nodes of V_1. The nodes in V_1 in turn influence the nodes in $V_2 \setminus S'$. Hence, S' is a feasible solution for the TSS Problem on G. This completes the proof.

3.1 Approximation hardness

In this section, we study the TSS Problem on bipartite graphs and obtain an $O(\log n_{min})$ factor inapproximability result by a reduction from the classical set cover problem, where n_{min} is the cardinality of the smaller part in the bipartition. First, we state the optimization version of the set cover problem.

> **Set Cover Problem (Optimization Version)**
>
> *Instance: A ground set $X = \{x_1, x_2, \ldots, x_n\}$ of n elements, a collection of m subsets $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ of X.*
>
> *Problem: Find a minimum-sized sub-collection $\mathcal{T}' \subseteq \mathcal{T}$ such that*
> $$\bigcup_{T \in \mathcal{T}'} T = X.$$

An incremental greedy approach, which starts with an empty set and in each iteration picks a subset that covers the maximum number of uncovered elements, yields an $\mathcal{H}_n = \frac{1}{n} + \frac{1}{n-1} + \cdots + 1 \simeq \log n$ factor approximation guarantee, and this bound is tight.

Theorem 4. *[13] Unless $P = NP$, the Set Cover Problem cannot have a polynomial-time approximation algorithm with a performance guarantee better than $O(\log n)$.*

Next, we state our reduction from the set cover problem to the TSS Problem.

Construction 1 *Let $(X, \mathcal{T})$ be an instance of the set cover problem, where $X = \{x_1, \ldots, x_n\}$ and $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a family of subsets of X. From this instance of the set cover problem, we construct an instance of the optimization version of the TSS problem on a bipartite graph denoted by (G, V_1, V_2, E, τ), where $V_1 = \{x_1, \ldots, x_n\}$ and $V_2 = \{T_1, \ldots, T_m\}$, $E = \{x_i T_j : x_i \in T_j, i \in [n], j \in [m]\}$ and $\tau(x_i) = 1$, $\forall i \in [n]$, $\tau(T_j) = |T_j|$, $\forall j \in [m]$.*

Remark 2. Given an instance $(X, \mathcal{T})$ of the set cover problem, let $\mathcal{S} \subseteq \mathcal{T}$ be a feasible set cover. Then, $\mathcal{S}$ is also a target set for the instance (G, V_1, V_2, E, τ) of the TSS Problem constructed from $(X, \mathcal{T})$ using Construction 1.

Remark 2 gives rise to the following remark.

Remark 3. Let (G, V_1, V_2, E, τ) be an instance of the TSS Problem constructed from an instance $(X, \mathcal{T})$ of the set cover problem using Construction 1. Then, the size of an optimal target set for (G, V_1, V_2, E, τ) is at most the size of an optimal set cover for $(X, \mathcal{T})$.

Theorem 5. *Unless $P = NP$, if the underlying influence graph is bipartite, the optimization version of the TSS Problem cannot have a polynomial time approximation algorithm with a performance guarantee better than $O(\log n_{min})$, where n_{min} is the cardinality of the smaller part in the bipartition.*

Proof. We prove this statement by an approximation preserving reduction from the set cover problem. Given an instance $(X, \mathcal{T})$ of the set cover problem with $X = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, we generate an instance (G, V_1, V_2, E, τ) of the TSS Problem on a bipartite graph as stated in Construction 1. Let us assume that $\mathcal{A}$ is a polynomial time algorithm for the TSS Problem having an approximation guarantee better than a factor of $O(\log n_{min})$, where n_{min} is the cardinality of the smaller part in the bipartition. We run the algorithm on the bipartite influence graph and obtain a seed set $S^{\mathcal{A}}$. Now, we construct a seed set S from $S^{\mathcal{A}}$ with $|S| \leq |S^{\mathcal{A}}|$ that satisfies the two properties given in Lemma 2. We know from Lemma 2 that such an S can be constructed in polynomial time. Let $S = \{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$. Since S satisfies Property (ii) of Lemma 2, it is a valid set cover of $(X, \mathcal{T})$. From Remark 3, we know that S is a set cover of $(X, \mathcal{T})$ of size less than $O(\log n)$ times the size of a minimum set cover. Hence, Algorithm $\mathcal{A}$ combined with Construction 1 can be used to solve the set cover problem with an approximation guarantee better than a factor of $O(\log n)$, which according to Theorem 4 is not possible unless $P = NP$.

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. Bazgan, C., Chopin, M., Nichterlein, A., Sikora, F.: Parameterized inapproximability of target set selection and generalizations. Computability 3(2), 135–145 (2014)
3. Bliznets, I., Sagunov, D.: Solving target set selection with bounded thresholds faster than 2^n. arXiv preprint arXiv:1807.10789 (2018)
4. Chen, N.: On the approximability of influence in social networks. SIAM Journal on Discrete Mathematics 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. Journal of Combinatorial Optimization 25(4), 702–715 (2013)
6. Chiang, C.Y., Huang, L.H., Yeh, H.G.: Target set selection problem for honeycomb networks. SIAM Journal on Discrete Mathematics 27(1), 310–328 (2013)
7. Chierichetti, F., Lattanzi, S., Panconesi, A.: Rumor spreading in social networks. Theoretical Computer Science 412(24), 2602–2610 (2011)
8. Chopin, M., Nichterlein, A., Niedermeier, R., Weller, M.: Constant thresholds can make target set selection tractable. Theory of Computing Systems 55(1), 61–83 (2014)
9. Dvořák, P., Knop, D., Toufar, T.: Target set selection in dense graph classes. arXiv preprint arXiv:1610.07530 (2016)
10. Hu, Y.C., Perrig, A., Johnson, D.B.: Wormhole attacks in wireless networks. IEEE journal on selected areas in communications 24(2), 370–380 (2006)
11. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. Social Network Analysis and Mining 3(2), 233–256 (2013)
12. 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 p. 201009094 (2010)
13. Vazirani, V.V.: Approximation algorithms. Springer Science & Business Media (2013)