Threshold-Bounded Dominating Set with Incentives

Gennaro Cordasco[1], Luisa Gargano[2], and Adele A. Rescigno[2]

[1] Università della Campania "Luigi Vanvitelli", Caserta, Italy
[2] Department of Computer Science, Università di Salerno, Italy

Abstract. Motivated by some problems in the area of influence spread in social networks, we introduce a new variation on the domination problem which we call *Threshold-Bounded Domination with Incentives*. Let $G = (V, E)$ be a graph with an influence threshold $t(v)$ for each node. An assignment of external incentives to the nodes of G is a cost function $c : V \to \mathbb{N}_0$, where $c(v)$ is the incentive given to $v \in V$. The effect of applying incentive $c(v)$ to node v is to decrease its threshold, i.e., to make v more susceptible to be dominated. A node is in the Threshold-Bounded dominating set D if it receives an incentive equal to its threshold, that is, $c(v) = t(v)$. A node, which is not in D, is dominated if the number of its neighbors in D plus the incentives it has received is at least equal to the node threshold $t(v)$. The goal is to minimize the total of the incentives required to ensure that all the nodes are dominated. The problem is log-APX-complete in general networks with unbounded degree. We prove that a greedy strategy has approximation factor $\ln \Delta + 2$, where Δ is the maximum degree of a node. We also give exact linear time algorithms for some classes of graphs.
Keywords: Weighted domination, Vector domination, Approximation algorithms, Recommending systems, Social networks, Incentives.

1 Introduction

Let $G = (V, E)$ be a graph modeling a network. For any node $v \in V$, we denote by $N(v) = \{w \mid (v, w) \in E\}$ and by $d(v) = |N(v)|$, respectively, the neighborhood and the degree of the node v in G. We denote by $n = |V|$ and $\Delta = \max_{v \in V} d(v)$ the number of nodes and the maximum degree of G respectively. Moreover, for each set $D \subseteq V$, we denote by $N_D(v) = N(v) \cap D$, the neighbors of v belonging to D and by $d_D(v) = |N_D(v)|$ its cardinality. A Dominating Set for $G = (V, E)$ is a subset $D \subseteq V$ of nodes such that $N_D(v) \neq \emptyset$, for each $v \in V - D$.

The concept of domination in graphs has been intensively studied, under several variations of dominating sets in graphs, because of its various applications in several research areas [17, 18]. In this paper we introduce and study a novel domination problem which we call *Threshold-Bounded Domination with Incentives*.

1.1 The Problem

Let $t : V \to \mathbb{N} = \{1, 2, \ldots\}$ be a function assigning integer thresholds to the nodes of G. An assignment of incentives to the nodes of a network $G = (V, E)$ is a cost function $c : V \to \mathbb{N}_0 = \{0, 1, 2, \ldots\}$, where $c(v)$ is the incentive given to $v \in V$. The effect of applying the incentive $c(v)$ to node v is to decrease its threshold, i.e., to make v more susceptible to be dominated. A node is in the *Threshold-Bounded dominating set* D if it receives an incentive equal to its threshold, that is, $c(v) = t(v)$. A node in $V - D$ is *dominated* if the number of its neighbors in D plus the incentives it has received is at least equal to the node threshold $t(v)$. We assume, w.l.o.g., that[3] $0 \leq c(v) \leq t(v) \leq d(v)$.

(otherwise, we can set t(u)=d(u)+1 for every node u with threshold exceeding its degree plus one without changing the problem

Given a set $D \subseteq V$, consider the incentive function $c : V \to \mathbb{N}_0$ with

$$c(v) = \begin{cases} t(v) & if v \in D \\ \max\{t(v) - d_D(v),\, 0\} & if v \in V - D. \end{cases}$$

We say that D is a *Threshold-Bounded (TB) dominating set* with incentives of cost $\sum_{v \in V} c(v)$. We are interested in finding the TB dominating set of minimum cost. Formally, we study the following problem.

THRESHOLD-BOUNDED DOMINATING SET WITH INCENTIVES (TBDI).

Instance: A network $G = (V, E)$ with thresholds $t : V \longrightarrow \mathbb{N}$.

Problem: Find a set $D \subseteq V$ of nodes which minimizes the following value.

$$c(D) = \sum_{v \in D} t(v) + \sum_{\substack{v \in V-D \\ t(v) > d_D(v)}} (t(v) - d_D(v)). \qquad (1)$$

In the following we will refer to the set $D \subseteq V$ that minimizes (1) as an *optimal TB dominating set in G*.[4]

1.2 Related work

Domination in Graphs. Domination problems in graphs have been extensively studied under various assumptions, both from structural and algorithmic points of view. We list here some of them which are close to the one considered in this paper. We assume the graph be $G = (V, E)$.

α-domination. Close related to our work is the concept of α-domination introduced in [10] by Dunbar *et al.*. A set $D \subseteq V$ is an α-dominating set of G, if $d_D(u) \geq \alpha d(u)$ for all $u \in V - D$, i.e., for every node that does not belong to D at least an α-fraction of its neighbors belong to D. The goal is to minimize the size of D.

Monopolies. A set $M \subseteq V$ is called a self-ignoring r-monopoly if every $v \in V - M$ is dominated r times, that is, $|N_M(v)| \geq r$. The goal is to minimize the size of M. Monopolies have been widely studied in the literature, see [24] for a survey.

Vector Domination/Threshold Ordinary Domination/f-Domination. Given a vector $k = (k_v | v \in V)$ with $k_v \leq d(v)$ for each $v \in V$, the vector domination problem asks to find a vector dominating set (VDS) of minimum size, that is, a set S minimizing $|S|$ and such that $|N_S(v)| \geq k_v$ for all $v \in V$ [16, 1]. This problem also appeared in the literature under the name of threshold ordinary dominating sets [15] and f-domination [26]; it also coincides with Threshold-Bounded Influence Dominating Set in [12].

[3] Notice that in the case $t(v) > d(v)$, we can set $t(v) = d(v)$, while the exceeding value $t(v) - d(v)$ will be added to $c(v)$.

[4] Notice that the optimal set can always be chosen as a dominating set of the input graph; indeed if D does not dominate a node $v \in V$, then applying (1) to the sets D and $D' = D \cup \{v\}$, one gets $c(D) - c(D') = \sum_{u \in N(v)} \max\{0, d_D(u) - t(u)\} \geq 0$.

Minimum-weighted dominating set. The Minimum-Weighted Dominating Set problem (MWDS) is a generalization of the Dominating Set, which asks to find a Dominating Set of a graph G of minimum total weight. Until now, the best known approximation ratio for an MWDS in a general graph is $O(\log \Delta)$.

We stress that in all the above unweighted problems the focus is on the minimization of the size of the dominating set attaining the desired property. In case of weighted networks, the focus is on the minimization of the total weight of the set. However, the solution of the TBDI problem can be quite different from the solutions of the weighted Vector Dominating Set problem for a given network. As an example, consider a complete graph K_8 on 8 nodes all with threshold/weight 7. The optimal solution for the weighted Vector Dominating Set problem consists of any subset of 7 nodes and its cost is $7 \times 7 = 49$. The TBDI problem admits a different optimal solution where the set D consists of 4 nodes and the remaining nodes need an incentive of value 3. The overall cost of the incentives is therefore $7 \times 4 + 3 \times 4 = 40$.

Social Networks. In addition to its theoretical interest, the study of the TBDI problem is also relevant to the area of influence propagation in social networks.

The problem of identifying influential individuals in a social network received a lot of attention, as it has applications in several areas, including viral marketing, disease prevention, disease propagation, politics, etc., see [19] for a recent survey. For instance, in the area of recommendation systems [11], companies wanting to promote products might try initially to convince a few individuals (using discounts for instance) which, exploiting social contagion, can trigger a cascade of influence in the network, leading to an adoption of the products by a much larger number of individuals.

In this paper, we use the well-known threshold model to study the influence diffusion process in social networks: For each node $v \in V$, the threshold value $t(v)$ quantifies how hard/easy it is to influence v [14]. Most of the existing approaches regard this as a long-term diffusion process, where information cascading occurs in several rounds [4, 5, 8]. Such models do not fit situations where time is an issue. In practice, time constraints and spreading speed are always critical concerns of marketers since they are closely related to profit and competition [21, 22]. The time problem in the spreading process has been recently algorithmically addressed in [3, 2, 12]. In particular, one round influence propagation has been recently studied in [12] as a building block for the design of an effective recommendation system.

The classical model, also adopted in [12], limits the optimizer to a binary choice between zero or complete initial influence on each individual. This implies that the cost of influencing a very popular (and therefore highly influential) node is the same as that of influencing a lonely individual and only the number of nodes in the dominating set is considered as a cost measure. Moreover, customized incentives (such as free copies, discounts, rewards, . . .) that can be given to individual nodes could be more effective in realistic scenarios [7, 9, 20, 23].

The proposed TBDI problem, in which the cost of totally influence a node is proportional to the node threshold (that can in turn be proportional to the node influence capability) and a partial external influence can be applied to each node, can then be used, among the other possible applications, to model a recommendation system (cfr. [12]) with a realistic cost counting. The above issues have been considered in the recent paper

[6] dealing with the use of external incentives in time-bounded influence spreading: Let λ be a bound on the number of rounds available to complete the process of influencing all nodes of the network, find incentives of minimum cost which result in all nodes being influenced in at most λ rounds. Formally, an influence process in $G = (V, E)$, with node thresholds $t : V \to \mathbb{N}$, starting with incentives in $c : V \longrightarrow \mathbb{N}_0$ is a sequence of node subsets $I_0 = \{v \,|c(v) = t(v)\}$ and $I_\ell = I_{\ell-1} \cup \left\{v \,\middle|\, \left|N_{I_{\ell-1}}(v)\right| \geq t(v) - c(v)\right\}$, for $\ell \geq 1$. The TIME-BOUNDED TARGETING WITH INCENTIVES problem studied in [6] takes in input a network $G = (V, E)$ with node thresholds $t : V \to \mathbb{N}$ and a time bound λ and asks for an incentive assignment $c : V \longrightarrow \mathbb{N}_0$ of minimum cost $\sum_{v \in V} c(v)$ such that $I_\lambda = V$.
It has been shown in [6] that the above problem, in general graphs, cannot approximated to within a ratio of $O(2^{\log^{1-\epsilon} n})$, for any fixed $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$, while exact algorithms are given for paths, complete graphs and trees. Specifically, the complexity of the presented algorithm is $O(n)$ for paths, $O(\lambda n \log n)$ for complete networks and $O(\lambda^2 \Delta n)$ for trees.
The TBDI problem considered in this paper coincides with the case in which only one round of influence can be assumed. Indeed our problem can be seen as to find $c : V \longrightarrow \mathbb{N}_0$ of minimum cost $\sum_{v \in V} c(v)$ such that $D = I_0$ and $I_1 = V$. In such a case, we can get novel and stronger results as described in Section 1.3.

1.3 Our Results

Our main contributions are the following: In Section 2, we prove that a natural greedy strategy has approximation factor $\ln \Delta + 2$. Unless $P \approx NP$, the approximation ratio of the greedy algorithm is optimal (up to lower order terms), since the TBDI problem is not easier than the dominating set problem and, consequently, cannot be approximated to within a ratio better than $(1 - o(1)) \times \ln \Delta$, unless $NP \subseteq DTIME(n^{polylog(n)})$. In Section 3, we give exact *linear-time* algorithms for complete and tree networks.

2 General Graphs

We show in this section that the TBDI problem can be optimally approximated within a logarithmic factor.

We first notice that the TBDI problem defined in Section 1.1 includes the dominating set (*DS*) problem (in case each node has threshold 1). It was shown in [25] that unless $NP \subseteq DTIME(n^{O(\log \log n)})$, no polynomial-time algorithm can approximate the DS problem better than $(1 - o(1)) \times \ln \Delta$.

Theorem 1. *The TBDI problem cannot be approximated to within a ratio better than* $(1 - o(1)) \times \ln \Delta$*, unless* $NP \subseteq DTIME(n^{polylog(n)})$.

We show now that the logarithmic factor approximation for the TBDI problem can be attained by a greedy algorithm.

In the following, let $D \subseteq V$ be a TB dominating set, we name *black* the nodes that belong to the set D. Moreover, for each node $v \in V - D$ we denote by

$$t'(v) = \max\{t(v) - d_D(v), 0\} \tag{2}$$

the residual threshold of the node v, that is the amount of incentive that still is required to dominate v using D as TB dominating set. Accordingly, we partition the non-black nodes in $V - D$ into two sets:

Algorithm 1: Greedy-TBDI(G)

Input: A graph $G = (V, E)$ with thresholds $t : V \longrightarrow \mathbb{N}$.
Result: A TB dominating set $D \subseteq V$ of minimum cost $c(D)$.

$D = \emptyset, W = V$
forall the $v \in V$ **do** $t'(v) = t(v)$
while $\exists u \in V - D$ *such that* $t(u) < w(u)$ **do** // Select v with largest $w(v)/t(v)$.
  $v = argmax_{u \in V-D} \left(\frac{w(u)}{t(u)}\right)$
  $D = D \cup \{v\}$
  $W = W - \{v\}$
  forall the $u \in N_W(v)$ **do**
    $t'(u) = t'(u) - 1$
    if $t'(u) = 0$ **then** $W = W - \{u\}$
return D

• the set W of *white* nodes, which contains non-dominated nodes ($t'(v) > 0$);
• the set of *gray* nodes, which contains dominated nodes ($t'(v) = 0$).

Finally, we denote by $w(v)$ the span of the node v, which corresponds to the number of white neighbours plus the residual threshold ($w(v) = d_W(v) + t'(v)$).

Algorithm 1 greedily selects the node with the largest value $w(v)/t(v)$ until there is a node having threshold smaller that its span. It is possible to see that the algorithm can be implemented in such a way to run in $O(|E| \log |V|)$ time. Indeed we need to process the nodes $v \in V$ according to the metric $w(v)/t(v)$, and the updates that follow each processed node $v \in V$ involve at most the $d(v)$ neighbors of v.

Theorem 2. *Algorithm 1 is a* $(\ln \Delta + 2)$*-approximation for the TBDI problem.*

Proof. (Sketch)

We first show that the TBDI problem is a special case of the following WTVD problem[5]:

WEIGHTED TARGETED VECTOR DOMINATION (WTVD).
Input: A graph $G = (V, E)$, thresholds $t : V \longrightarrow \mathbb{N}$, and a target set $U \subseteq V$.
Question: Find a set $D \subseteq V$ of minimum cost $\sum_{v \in D} t(v)$ such that $d_D(u) \geq t(u)$, for each node $u \in U - D$.

It is possible to see the existence of a reduction from TBDI to WTVD as follows. Let $G = (V, E)$ with thresholds $t : V \longrightarrow \mathbb{N}$ be an instance of the TBDI problem. For each node v having threshold $t(v)$, we add to the network a set $\{v_1, v_2, \ldots, v_{t(v)}\}$ of $t(v)$ nodes having threshold 1, connected to the node v and we obtain an new graph $H = (V(H), E(H))$ with thresholds $t_H : V(H) \longrightarrow \mathbb{N}$ (see Fig. 1) where:

- $V(H) = U \cup U'$, where $U = V$ and $U' = \bigcup_{v \in V} \{v_1, v_2, \ldots, v_{t(v)}\}$,
- $E(H) = E \cup E'$, where $E' = \bigcup_{v \in V} \{(v, v_i),\ \text{for } i = 1, 2, \ldots, t(v)\}$,
- $t_H(v) = \begin{cases} t(v) & v \in U; \\ 1 & \text{otherwise.} \end{cases}$

[5] To the best of our knowledge, this is a novel domination problem which can be of interest on its own.

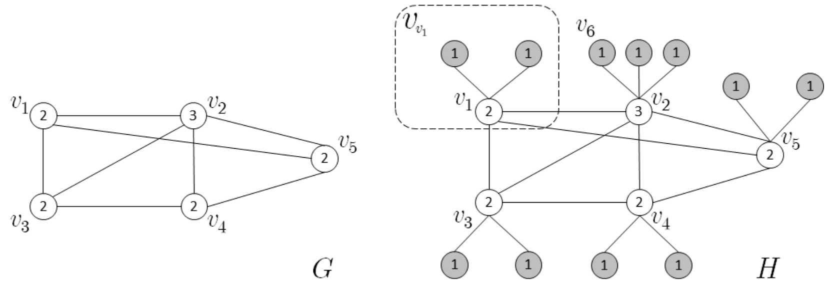


Fig. 1. The graph H (right) obtained starting from the graph G (left). The integers inside circles are the node thresholds. The dashed roundbox represents the gadget associated to the node v_1.

Claim 1 *There exists a TB dominating set $D \subseteq V$ in G such that $c(D) \leq k$ iff there exists a set $D_H \subseteq V(H)$ in H such that $\sum_{v \in D_H} t_H(v) \leq k$ and $d_{D_H}(u) \geq t_H(u)$, for each $u \in U - D_H$.*

Algorithm 2 provides a solution D_H for the WTVD problem having the same cost of the solution for the TBDI problem. Indeed, the solution for the TBDI problem can be obtained as $D = D_H \cap V$. It is worth observing that, since the nodes in $V - U$ are set as gray at the beginning of the Algorithm 2 (see line 3), both Algorithms 1 and 2 select the same nodes from the set $V = U$ as long as there exists a node $u \in V - D$ such that $t(u) < w(u)$ (each node $v \in U'$ has $t(v) = 1$ and $w(v) \leq 1$). Then Algorithm 1 ends with the set D, while Algorithm 2 continues selecting some nodes in U' in order to dominate all the white nodes. As observed above, the cost of the nodes selected in the set U' corresponds to the incentives applied to v for the TBDI problem.

Table 1 shows an example of the execution of the Algorithms 1 and 2 on the graph G and H in Fig. 1. For each iteration the set D (TB dominating set), the set W (white nodes) and for each node u, the metric $w(u)/t(u)$ are provided. The Algorithm 1 ends after 2 iterations while the Algorithm 2 needs 3 iterations. We are now going to show that the cost of the solution D_H provided by Algorithm 2 is a $(\ln \Delta + 2)$-approximation compared to the cost of an optimal solution D^* that is

$$\frac{\sum_{v \in D_H} t(v)}{\sum_{v \in D^*} t(v)} \leq \ln \Delta + 2. \tag{3}$$

Each time, the Algorithm 2 selects a new node v of the dominating set (each greedy step), we have cost $t(v)$. Instead of letting this node pay the whole cost, we distribute

step	D	D_H	W	$w(u)/t(u)$							v	$D \cup \{v\}$	$D_H \cup \{v\}$
				$V = U$					U'				
				v_1	v_2	v_3	v_4	v_5	v_6	$\dots$			
1	$\emptyset$	$\emptyset$	$\{v_1, v_2, v_3, v_4, v_5\}$	5/2	7/3	5/2	5/2	5/2	1	$\dots$	v_1	$\{v_1\}$	$\{v_1\}$
2	$\{v_1\}$	$\{v_1\}$	$\{v_2, v_3, v_4, v_5\}$	–	5/3	3/2	5/2	3/2	1	$\dots$	v_4	$\{v_1, v_4\}$	$\{v_1, v_4\}$
3	–	$\{v_1, v_4\}$	$\{v_2\}$	–	1/3	1/2	–	1/2	1	$\dots$	v_6	–	$\{v_1, v_4, v_6\}$

Table 1. The execution of the Algorithms 1 and 2 on the graph G and H depicted in Fig. 1.

Algorithm 2: Greedy-WTVD(G)

Input: A graph $G = (V, E)$ with thresholds $t : V \longrightarrow \mathbb{N}$ and a set $U \subseteq V$.
Result: A set $D_H \subseteq V$ of minimum cost $\sum_{v \in D_H} t(v)$ s.t. $\forall v \in U - D_H, d_{D_H}(v) \geq t(v)$.

1 $D_H = \emptyset, W = U$ // At the beginning, nodes in U are white.
2 **forall the** $v \in U$ **do** $t'(v) = t(v)$
3 **forall the** $v \in V - U$ **do** $t'(v) = 0$ // At the beginning, nodes in $V - U$ are gray.
4 **while** $\exists u \in W$ **do** // Select the node with the largest value $w(v)/t(v)$.
5 $\quad v = argmax_{u \in V - D_H} \left(\frac{w(u)}{t(u)} \right)$
6 $\quad D_H = D_H \cup \{v\}$
7 $\quad W = W - \{v\}$
8 $\quad$ **forall the** $u \in N_W(v)$ **do**
9 $\quad\quad t'(u) = t'(u) - 1$
10 $\quad\quad$ **if** $t'(u) = 0$ **then** $W = W - \{u\}$
11 **return** D_H

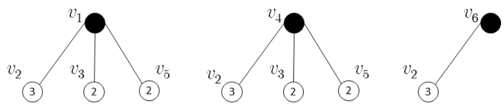


Fig. 2. The stars associated to the solution $D^* = \{v_1, v_4, v_6\}$ for the graph H depicted in Fig. 1.

the cost to all the nodes affected by this choice, plus the node v itself (when $t'(v) > 0$) according to the portion of the span $w(v)$ covered by each node. For instance consider the node v_1 in Fig. 1 having initial threshold $t(v_1) = 2$:
- Assume that v_1, is selected in line 5 of the Algorithm 2 when its residual threshold is $t'(v_1) = 2$ (i.e., as a white node) and it has three white neighbors. In this case, the span of the node v_1 is $w(v_1) = d_W(v_1) + t'(v_1) = 5$ and each of its white neighbors gets charged $t(v_1)/w(v_1) = 2/5$, while v_1 is charged $(t(v_1) \times t'(v_1))/w(v_1) = 4/5$. Overall the charge is $(t(v_1) \times d_W(v_1))/w(v_1) + (t(v_1) \times t'(v_1))/w(v_1) = 2 = t(v_1)$.
- On the other hand, if v_1 is selected as a gray node ($t'(v_1) = 0$), while it has two white neighbors, only the neighbors get charged (they all get $t(v_1)/w(v_1) = 2/2 = 1$) and again the overall charge is $t(v_1)$.

Assume now that we know a dominating set D^* of minimum cost $\sum_{v^* \in D^*} t(v^*)$. By the definition of the WTVD, each node v, which is not in $U - D^*$ has at least $t(v)$ neighbor from D^*. Hence we can associate to each node $v \in U - D^*$, $t(v)$ nodes in D^*. Then for each node $v^* \in D^*$ we build a star, centered in v^*, which contains as leaves, the nodes in $U - D^*$ that have been associated to it. Hence we have $|D^*|$ stars, each having a dominator ($v^* \in D^*$) as center and non-dominators as leaves. An example is presented in Fig. 2. Notice that each node $v \in U - D^*$ appears in $t(v)$ stars (that is we have $t(v)$ copies of the node v). Clearly, the contribution of the star centered in v^* to the overall cost of an optimal solution is $t(v^*)$.

It is possible to show that the amortized cost (distributed costs) of the Algorithm 2 is at most $t(v^*)(\ln \Delta + 2)$ for each star. This suffices to prove (3). □

3 Efficient algorithms for the TBDI problem

In this section we show that the TBDI problem can be efficiently solved when the graph G is a complete graph or a tree. Our results improve on those in [6] where polynomial time exact algorithm were presented for the Time-Bounded Targeting with Incentives problem within a fixed number $\lambda \geq 1$ of rounds. For the TBDI problem we show the existence of *linear* time algorithms.

3.1 Complete graphs

Let $K_n = (V, E)$ be the complete graph with node set $V = \{v_1, v_2, \ldots, v_n\}$. We will prove the following theorem.

Theorem 3. *For any complete network $K_n = (V, E)$, threshold function $t : V \longrightarrow \mathbb{N}$, the TBDI problem can be solved in linear time.*

The following lemma, proved in [6], gives us a tool to design our efficient algorithm.

Lemma 1. **[6]** *Given thresholds $t(v_1) \leq t(v_2) \leq \ldots \leq t(v_n)$, if there exists an optimal TB dominating set D^* in K_n such that $|D^*| = j$ then also $D = \{v_1, \ldots, v_j\}$ is an optimal TB dominating set in K_n.*

In the following, we assume $t(v_1) \leq t(v_2) \leq \ldots \leq t(v_n)$. Notice that the sorting can be done in $O(n)$ time using counting sort because $1 \leq t(v) \leq n - 1 = d(v)$ for all $v \in V$.

If the TB dominating set of K_n has j nodes, for some $1 \leq j \leq n$, then they are $v_1, v_2, \ldots, v_j$ by Lemma 1; furthermore, each of the $n - j$ remaining nodes in V has j neighbors (i.e., $v_1, v_2, \ldots, v_j$) in the TB dominating set. For such a remaining node v either $t(v) \leq j$ or v has incentive $c(v) = t(v) - j$. Hence, if we denote by $D_j = \{v_1, v_2, \ldots, v_j\}$, it follows that the optimal TB dominating set of K_n is

$$D = \arg\min_{1 \leq j \leq n} c(D_j) = \arg\min_{1 \leq j \leq n} \left(\sum_{i=1}^{j} t(v_i) + \sum_{i=j+1}^{n} \max\{t(v_i) - j, 0\} \right). \quad (4)$$

We show now that D can be computed in time $O(n)$. To this aim, we can pre-compute an auxiliary vector s where, for $j = 1, \ldots, n$, $s[j]$ contains the smallest index $i \geq j$ such that $t(v_i) > j$ if it exists, otherwise $s[j] = n+1$. Clearly, $s[1] \leq s[2] \leq \ldots \leq s[n]$. The vector s can be precomputed in $O(n)$ time once the nodes are sorted by non-decreasing threshold value. Noticing that

$$c(D_1) = t(v_1) + \sum_{i=s[1]}^{n} (t(v_i) - 1), \quad \text{and for } j = 2, \ldots, n$$

$$\begin{aligned}
c(D_j) &= \sum_{i=1}^{j} t(v_i) + \sum_{i=s[j]}^{n} (t(v_i) - j) \\
&= \sum_{i=1}^{j} t(v_i) + \sum_{i=s[j-1]}^{n} (t(v_i) - j + 1 - 1) - \sum_{i=s[j-1]}^{s[j]-1} t(v_i) - j \\
&= c(D_{j-1}) + \left(t(v_j) - (n - s[j-1] + 1) - \sum_{i=s[j-1]}^{s[j]-1} t(v_i) - j \right)
\end{aligned}$$

we obtain that $O(n)$ total time suffices to compute all the $c(D_j)$, for $j = 1, \ldots, n$.

Example 1. Let $K_8 = (V, E)$ be a complete graph having 8 nodes $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ having thresholds $1, 2, 3, 3, 5, 5, 5, 5$ respectively. Using the above strategy, we have that an optimal TB dominating set is $D^* = \{v_1, v_2, v_3, v_4\}$. Hence, the optimal incentive function is $c(v) = t(v)$, for each $v \in D^*$, while, for each $v \notin D^*$, $c(v)$=5$-|D^*|$=1 and $c(D^*) = \sum_{v \in D^*} t(v) + \sum_{v \notin D^*} c(v) = 9 + 4 = 13$.

3.2 Trees

Let $T = (V, E)$ be a tree having n nodes. In the following, we will assume that T is rooted at some node r and for any node v, we denote by T_v the subtree rooted at v, by $C(v)$ the set of children of v and, whenever v is not the root r, by $p(v)$ the parent of v. Furthermore, for any incentive function c and any node v, we will denote by $c(T_v)$ the sum of the incentives c assigns to the nodes in T_v.

We give a dynamic programming algorithm that proves Theorem 4.

Theorem 4. *For any tree $T = (V, E)$, threshold function $t : V \longrightarrow \mathbb{N}$, the TBDI problem can be solved in linear time.*

The algorithm performs a post-order visit of T — so that each node is considered after all of its children have been processed — and for each node v, it solves three different TBDI problems on the subtree T_v, according to some conditions on v regarding the fact that it and its parent belong or not in the TB dominating set.

Definition 1. *Let D be a TB dominating set in T. For each $v \in V$, we define:*
- $Y[v]$ *as the minimum cost for TB dominating all the nodes in T_v, in case $v \in D$.*
- $N[v]$ *as the minimum cost for TB dominating all the nodes in T_v, assuming that both $v \notin D$ and $p(v) \notin D$.*
- $R[v]$*, if $v \neq r$, as the minimum cost for TB dominating all the nodes in T_v, assuming that $v \notin D$ and $p(v) \in D$.*

Notice that if $p(v) \in D$ then the threshold of node v in T_v is $t(v) - 1$ since $p(v)$ is one of the neighbors of v in D.

We set the above values equal to infinity when their constraints are not satisfiable. For instance for a leaf node v if $v \notin D$ and $p(v) \notin D$ then v can not be dominated and consequently $N[v] = \infty$.

The minimum cost of the TB dominating set in T can then be obtained by computing

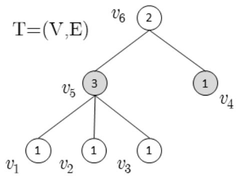
$$\min\{Y[r], N[r]\}. \tag{5}$$

We proceed in a post-order fashion, so that the computations of the values $Y[v]$, $N[v]$ and $R[v]$ for a node v are done after all of the values for v's children are known.

Consider a leaf v (with threshold $t(v) = 1$) and the one-node subtree T_v. Either v is in the TB dominating set or its parent $p(v)$ is in the dominating set (i.e., the threshold of v in T_v is $t(v) - 1 = 0$), we have that for each leaf node v holds

$$Y[v] = t(v) = 1 \qquad N[v] = \infty \qquad R[v] = t(v) - 1 = 0 \tag{6}$$

Let v be any internal node. In order to compute the value $Y[v]$, consider that we assume that v is in the TB dominating set and then the incentive of v is exactly its



	v_1	v_2	v_3	v_4	v_5	v_6
Y[v]	1	1	1	1	3	5
N[v]	∞	∞	∞	∞	3	4
R[v]	0	0	0	0	3	-

Fig. 3. An example of three with 6 nodes. The value inside the circle denotes the node threshold. Gray nodes represents the nodes in the optimal TB dominating set. For each node v, the values $Y[v]$, $N[v]$ and $R[v]$ are presented in the table.

threshold $t(v)$; furthermore, the minimum cost for TB dominating all the nodes in the subtree rooted at any children $u \in C(v)$ is the minimum between $Y[u]$ (that reflects the case that u belong to the TB dominating set), and $R[u]$ (that reflects the fact that u is not in the TB dominating set while v (the parent of u) is in the TB dominating set and then the threshold of u is $t(u) - 1$). It follows that $Y[v]$can be computed in $O(d(v))$ time as

$$Y[v] = t(v) + \sum_{u \in C(v)} \min\{Y[u], R[u]\}. \tag{7}$$

Now, we derive the value $N[v]$. Recalling that in this case v and its parent are not in the TB dominating set we have that its threshold is $t(v)$ and the threshold of each children $u \in C(v)$ is $t(u)$. In this case at least one of its children should be in the TB dominating set. Let S be a subset of $j \geq 1$ children of v that are in the TB dominating set. Hence, $t(v) - j$ is the incentive of v. Furthermore, the minimum cost for TB dominating all the nodes in each subtree T_u with $u \in S$ is $Y[u]$, while the minimum cost for TB dominating all the nodes in each subtree T_u with $u \in C(u) - S$ is $N[u]$. It follows that

$$N[v] = \min_{1 \leq j \leq t(v)} \min_{\substack{S \subseteq C(v) \\ |S|=j}} \left(t(v) - j + \sum_{u \in S} Y[u] + \sum_{u \in C(v)-S} \min\{Y[u], N[u]\} \right) \tag{8}$$

The value $R(v)$ can be derived following the same arguments used to have $N[v]$ excepting for the fact that the threshold of v is $t(v) - 1$ and so the parent of v is a neighbor of v in the TB dominating set. This implies that if S is a subset of j children of v that are in the TB dominating set, it holds $0 \leq j \leq t(v) - 1$. It follows that

$$R[v] = \min_{0 \leq j \leq t(v)-1} \min_{\substack{S \subseteq C(v) \\ |S|=j}} \left(t(v) - 1 - j + \sum_{u \in S} Y[u] + \sum_{u \in C(v)-S} \min\{Y[u], N[u]\} \right) \tag{9}$$

Example 2. Let $T = (V, E)$ be the three in Fig 3. Using the above strategy, one can easily fill the table in the figure. The minimum cost of the TB dominating set in T corresponds to $N[v_6] = 4$. Indeed an optimal TB dominating set is $D^* = \{v_4, v_5\}$.

The time complexity of the computation of $N[v]$ and $R[v]$ can be strongly reduced by using a particular ordering of the children of v. Assume to have sorted the $d = d(v) - 1$ children of v, let say $v_1, v_2, \ldots, v_d$, according to the non-decreasing order of the differences between $Y[v_i]$ and $N[v_i]$ for each $i = 1, \cdots, d$, i.e.,

$$Y[v_1] - N[v_1] \ \leq \ Y[v_2] - N[v_2] \ \leq \cdots \leq \ Y[v_d] - N[v_d]. \tag{10}$$

In the following we prove that

$$-2\Delta \leq Y[v_i] - N[v_i] \leq \Delta \tag{11}$$

and then the sorting (10) can be done using counting sort in $O(\Delta)$ time. To have (11), we make two simple considerations:
– Take the incentives assigned to the nodes in T_{v_i} to have $N[v_i]$ and increase the incentive assigned to v_i so that it becomes $t(v_i)$. This gives a new solution for TB dominating all the nodes in T_{v_i} with v_i in the TB dominating set; hence, we have $Y[v_i] \leq t(v_i) + N[v_i]$ and so $Y[v_i] - N[v_i] \leq t(v_i) \leq d(v_i) \leq \Delta$.
– On the other hand, take the incentives assigned to the nodes in T_{v_i} to have $Y[v_i]$ and, decrease by 1 the incentive assigned to v_i, increase the incentive assigned to some $u \in C(v_i)$ so that it becomes $t(u) \leq d(u) \leq \Delta$, and finally increase by 1 the incentive assigned to each of the remaining $d(v_i) - 1$ children of v_i. This gives a new solution for TB dominating all the nodes in T_{v_i} in which v_i is not in the TB dominating set and whose cost is at most $Y[v_i]-1+d(v_i)-1+\Delta \leq Y[v_i]+2\Delta$ and so $Y[v_i]+2\Delta \geq N[v_i]$.

Lemma 2. *If there exists an optimal dominating set D^* in T_v such that $v \notin D^*$ and $j \geq 1$ children of v are in D^* then there exists an optimal dominating set D' in T_v such that $v \notin D'$ and $v_1, v_2, \cdots, v_j \in D'$.*

By using the ordering (10) and Lemma 2, the recurrence (8) can be simplified as follows.

$$N[v] = \min_{1 \leq j \leq t(v)} \left(t(v) - j + \sum_{i=1}^{j} Y[v_i] + \sum_{i=j+1}^{t(v)} \min\{Y[v_i], N[v_i]\} \right)$$

Hence, (8) can be computed in time $O(t)$. Similarly,

$$R[v] = \min_{0 \leq j \leq t(v)-1} \left(t(v) - 1 - j + \sum_{i=1}^{j} Y[v_i] + \sum_{i=j+1}^{t(v)} \min\{Y[v_i], N[v_i]\} \right)$$

can be computed in time $O(t)$.

In conclusion, the value in (5) can be computed in time

$$\sum_{v \in V} O(t(v)) \leq \sum_{v \in V} O(d(v)) = O(n).$$

Standard backtracking techniques can be used to compute an optimal TB dominating set in the same O(n) time.

References

1. F. Cicalese, M. Milanič, U. Vaccaro. On the approximability and exact algorithms for vector domination and related problems in graphs. *Discrete Applied Math.,* 161, (2013), 750–767.
2. F. Cicalese, G. Cordasco, L. Gargano, M. Milanič, U. Vaccaro. Latency-Bounded Target Set Selection in Social Networks. *Theoretical Computer Science, 535,* (2014), 1–15.
3. F. Cicalese, G. Cordasco, L. Gargano, M. Milanič, J. Peters, U. Vaccaro. Spread of Influence in Weighted Networks under Time and Budget Constraints. In *Theoretical Computer Science, 586,* (2015), 40–58.

4. G. Cordasco, L. Gargano, A.A. Rescigno. On finding small sets that influence large networks. In *Social Network Analysis and Mining, Vol. 6 (1),* (2016).
5. G. Cordasco, L. Gargano, A. A. Rescigno and U. Vaccaro Evangelism in social networks: Algorithms and complexity. In *Networks*, ISSN: 1097-0037, (2017).
6. G. Cordasco, L. Gargano, J. Peters, A. Rescigno, U. Vaccaro. Time-Bounded Influence Diffusion with Incentives, *Proc. of SIROCCO 2018.* (2018), (see also arXiv:1807.06921).
7. G. Cordasco, L. Gargano, M. Lafond, L. Narayanan, A. Rescigno, U. Vaccaro and K. Wu. Whom to befriend to influence people. In *Theoretical Computer Science (TCS)*, (2018).
8. G. Cordasco, L. Gargano, M. Mecchia, A.A. Rescigno, U. Vaccaro. Discovering Small Target Sets in Social Networks: A Fast and Effective Algorithm. In *Algorithmica, Vol. 535,* (2018).
9. E. D. Demaine, M.T. Hajiaghayi, H. Mahini, D. L. Malec, S. Raghavan, A. Sawant, M. Zadimoghadam. How to influence people with partial incentives. In *Proc. WWW '14*, (2014).
10. J.E. Dunbar, D.G. Hoffman, R.C. Laskar and L.R. Markus, α-domination, In *Discrete Math.* 211 (2000), 11–26.
11. D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, (2010).
12. M. Eirinaki, N. Moniz, K. Potika, *Threshold-Bounded Influence Dominating Sets for Recommendations in Social Networks*. In Proc. of IEEE International Conferences on Big Data and Cloud Computing, Social Computing and Networking, Sustainable Computing and Communications, (2016), 408–415.
13. L. Gargano, P. Hell, J. G. Peters, U. Vaccaro. Influence Diffusion in Social Networks under Time Window Constraints. In *Theoretical Computer Science (TCS) 584,* (2015), 53–66.
14. M. Granovetter. Thresholds Models of Collective Behaviors. *American Journal of Sociology*, Vol. 83, No. 6, (1978), 1420–1443.
15. W. Goddard, M.A. Henning. Restricted domination parameters in graphs *Journal of Combinatorial Optimization*, 13 (2007), 353–363.
16. J. Harant, A. Prochnewski, M. Voigt On dominating sets and independent sets of graphs Combinatorics, Probability and Computing, 8, (1999), 547–553.
17. T. W. Haynes, S. Hedetniemi, P. Slater, Fundamentals of Domination in Graphs *CRC Press*, (1998).
18. T. W. Haynes, S. Hedetniemi, P. Slater. Fundamentals of Domination in Graphs: Advanced Topics, *CRC Press*, (1998).
19. D. Kempe, J.M. Kleinberg, E. Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4),(2015), 105–147.
20. M. Lafond, L. Narayanan and K. Wu. Whom to Befriend to Influence People, *Proc. of SIROCCO 2016*, LNCS 9988, (2016), 340–357.
21. L. Liu, J. Tang, J. Han, S. Yang. Learning influence from heterogeneous social networks. *Data Min Knowl Disc* 25, (2012) 511–544.
22. A.J. Morales, J. Borondo, J.C. Losada, R.M. Benito. Efficiency of human activity on information spreading on twitter. In *Social Networks 39* , (2014), 1–11.
23. L. Narayanan, K. Wu. How to Choose Friends Strategically, In *Proc. of SIROCCO 2017*, LNCS 10641, (2017), 283–302.
24. D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science* 282, (2002), 231–257.
25. R. Raz and S. Safra. A Sub-constant Error-probability Low-degree Test, and a Sub-constant Error-probability PCP Characterization of NP. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing* (STOC '97), (1997), 475–484.
26. C. Stracke, L. Volkmann. A new domination conception. In *Journal of Graph Theory 17* (1993), 315–323.
27. M. Zaker. On dynamic monopolies of graphs with general thresholds. In *Discrete Mathematics*, 312(6), (2012), 1136-1143.