

The Complexity of Finding Harmless Individuals in Social Networks*

Cristina Bazgan^{1,2}

Morgan Chopin¹

1. PSL, Université Paris-Dauphine, LAMSADE UMR 7243
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France

2. Institut Universitaire de France
{bazgan,chopin}@lamsade.dauphine.fr

Abstract

In this paper, we introduce a domination-related problem called HARMLESS SET: given a graph $G = (V, E)$, a threshold function $t : V \rightarrow \mathbb{N}$ and an integer k , find a subset of vertices $V' \subseteq V$ of size at least k such that every vertex v in V has less than $t(v)$ neighbors in V' . We study its parameterized complexity and the approximation of the associated maximization problem. When the parameter is k , we show that the problem is W[2]-complete in general and W[1]-complete if all thresholds are bounded by a constant. Moreover, we prove that, if $P \neq NP$, the maximization version is not $n^{\frac{1}{2}-\varepsilon}$ -approximable for any $\varepsilon > 0$ even when all thresholds are at most two. When each threshold is equal to the degree of the vertex, we show that HARMLESS SET is fixed-parameter tractable for parameter k and the maximization version is APX-complete. We give a polynomial-time algorithm for graphs of bounded treewidth and a polynomial-time approximation scheme for planar graphs. Finally, we show that the parametric dual problem $(n - k)$ -HARMLESS SET is fixed-parameter tractable for a large family of threshold functions.

1 Introduction

The diffusion of information through social networks is a large and well-studied topic [20]. One of the most well known problems that appears in this context is TARGET SET SELECTION introduced by Chen [12] and defined as follows. The input is a graph where each vertex v has a threshold value $t(v)$, an integer k , and the following propagation rule: a vertex becomes active if at least $t(v)$ neighbors of v are active. The propagation process proceeds in several steps and stops when no further vertex becomes active. The task is then to determine the existence of a subset of at most k vertices, called a *target set*, such that all vertices of the input graph become active. This problem may occur for example in the context of disease propagation, faults in distributed computing or even viral marketing [19, 29, 36]. In this last example, the task for a company would be to advertise few but influential individuals such that by a so-called “word-of-mouth” process, a large fraction of customers is convinced about the usefulness of a product. This problem received considerable attention in a series of papers from classical complexity [10, 13, 19, 25, 37], polynomial-time approximability [1, 12],

*The preliminary results of this publication were presented at MFCS 2012 [4].

parameterized approximability [5], and parameterized complexity [6, 14, 33] perspectives. Altogether these results emphasize the strong intractability nature of this problem even on very restricted graph classes and threshold functions. A natural research direction considering this fact is to look for the complexity of variants or constrained versions of this problem. In this work, we follow this line of research by introducing the notion of *harmless set*. A harmless set consists of a set S of vertices with the property that no propagation occurs if any subset of S gets activated. In other words we define harmless set as a converse notion of target set. Formally, a set S is a harmless set if every vertex v of the input graph has less than $t(v)$ neighbors in S . Now we may ask for the complexity of the following HARMLESS SET problem. Given a graph $G = (V, E)$, a threshold function $t : V \rightarrow \mathbb{N}$, and an integer k , determine whether there exists a harmless set of size at least k (see Figure 1). Observe that in our definition of harmless set we impose the threshold condition on every vertex, including those in the solution S . Another perhaps more natural definition could have been a set S such that every vertex $v \notin S$ has less than $t(v)$ neighbors in S . This definition rises the following two problems. First, it makes HARMLESS SET meaningless as a trivial solution would be to take the whole set of vertices of the input graph. Second, there might be some propagation steps inside S if some vertices are activated in it.

Interestingly enough we may exhibit several connections between HARMLESS SET and other well known domination problems such as TOTAL DOMINATING SET [27] which is to find a set S of vertices such that every vertex of the input graph is adjacent to an element of S . One can observe that if all thresholds are unanimity (every threshold is equal to the degree of the vertex) then a harmless set is exactly the complement of a total dominating set also called a total non-blocker by Dehne et al. [16]. As a matter of fact, for the unanimity case, the relationship between a harmless set and a total dominating set is similar as the relationship between a non-blocker [16] or a spanning star forest [32] and a dominating set. While parameterized complexity and approximability results were found for the non-blocker or spanning star forest problems, such results are hitherto unknown for the complement version of the problem (that is finding a total non-blocker in a graph). This paper provides first results for this later problem. Furthermore, in the case of general thresholds, there exists an equivalence between our problem and a generalized version of TOTAL DOMINATING SET called ℓ -TUPLE TOTAL DOMINATING SET and introduced by Henning et al. [26]. The difference with the original version is that a vertex v is now dominated if and only if at least ℓ of its neighbors are in S . One can observe that an ℓ -tuple total dominating set is the complement of a harmless set when the threshold of every vertex v is set to $d(v) - \ell + 1$ where $d(v)$ is the degree of v . In a very recent survey, Fernau and Rodríguez-Velázquez provided a connection between harmless set and alliances in graphs [21]. Finally, we can relate our problem to (σ, ρ) -DOMINATING SET introduced by Telle [38]. Given a graph $G = (V, E)$, two sets σ, ρ of non-negative integers, and an integer k , the goal is to find an (σ, ρ) -dominating set $S \subseteq V$ of size at most k in G , *i.e.* $|S \cap N(v)| \in \sigma$ for every vertex $v \in S$ and $|S \cap N(v)| \in \rho$ for every vertex $v \notin S$. To see the relation between the two problems notice that if every threshold has the same value $c \in \mathbb{N}$ then HARMLESS SET is equivalent to finding an (σ, ρ) -dominating of size k [23] where $\sigma = \rho = \{0, \dots, c - 1\}$.

In this paper, we study the parameterized complexity of HARMLESS SET and the approximation of the associated maximization problem MAX HARMLESS SET. The paper is organized as follows. In Section 2 we give the definitions, terminology and preliminaries. In Section 3 we establish parameterized intractability results for HARMLESS SET with various threshold functions (see Table 1). We show that the parametric dual problem $(n - k)$ -HARMLESS SET

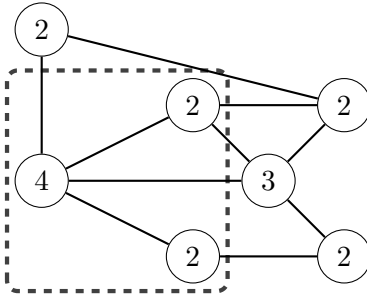


Figure 1: Example of a harmless set (dashed rectangle) of size three. The number in each vertex indicates the threshold value. Observe that there is no “propagation phenomenon” if one activates any vertices in the dashed rectangle.

is fixed-parameter tractable for a large family of threshold functions. In Section 4 we give a polynomial-time algorithm to solve HARMLESS SET for graphs of bounded treewidth. In Section 5 we establish that MAX HARMLESS SET is not $n^{\frac{1}{2}-\varepsilon}$ -approximable for any $\varepsilon > 0$ even when all thresholds are at most two. If each threshold is equal to the degree of the vertex, we show that MAX HARMLESS SET is APX-complete. Moreover MAX HARMLESS SET has a polynomial-time approximation scheme on planar graphs. Conclusions and open problems are given in Section 6.

Thresholds	HARMLESS SET	$(n - k)$ -HARMLESS SET	MAX HARMLESS SET
General	W[2]-complete (Th.4)	W[2]-hard	Not $n^{\frac{1}{2}-\varepsilon}$ -approx
Constant	W[1]-complete (Th.6)	FPT (Th.9)	Not $n^{\frac{1}{2}-\varepsilon}$ -approx (Th.13)
Majority	W[1]-hard (Th.6)	FPT (Th.9)	Not $n^{\frac{1}{2}-\varepsilon}$ -approx (Th.13)
Unanimity	FPT (Th.8)	W[2]-hard*	APX-complete (Th.15)

Table 1: A summary of our parameterized complexity results for HARMLESS SET and its parametric dual $(n - k)$ -HARMLESS SET along with our approximation results for MAX HARMLESS SET. Here, the parameter is k both for HARMLESS SET and $(n - k)$ -HARMLESS SET. The result marked with * is due to the equivalence between $(n - k)$ -HARMLESS SET with unanimity and the TOTAL DOMINATING SET problem proved W[2]-hard with respect to the solution size in [23].

2 Preliminaries

In this section, we give the notation used throughout this paper as well as the statement of the problems. We conclude by providing the basic background on parameterized complexity and approximation.

Graph terminology. Let $G = (V, E)$ be an *undirected graph*. The *neighborhood* of a vertex $v \in V$, denoted by $N_G(v)$, is the set of all neighbors of v . The *degree* of a vertex v is denoted by $d_G(v)$. We may simply write $N(v)$ and $d(v)$ if the graph is clear from the context.

Problem definitions. Let $G = (V, E)$ be an undirected graph, and $t : V \rightarrow \mathbb{N}$ a threshold function. A subset $V' \subseteq V$ is called *harmless* if for every $v \in V$ we have $|N(v) \cap V'| < t(v)$ (see Figure 1). In each figure, we indicate the thresholds inside the vertices. We define in the following the problems we study in this paper.

HARMLESS SET

Input: A graph $G = (V, E)$, a threshold function $t : V \rightarrow \mathbb{N}$ where $1 \leq t(v) \leq d(v)$ for every $v \in V$, and an integer k .

Parameter: k .

Question: Is there a harmless set $V' \subseteq V$ of size at least k ?

We also consider the parametric dual problem $(n - k)$ -HARMLESS SET which asks for the existence of a harmless set of size at least $n - k$. The parameter is still k and n denotes the number of vertices in the input graph.

The optimization version of HARMLESS SET is defined as follows.

MAX HARMLESS SET

Input: A graph $G = (V, E)$ and a threshold function $t : V \rightarrow \mathbb{N}$ where $1 \leq t(v) \leq d(v)$ for every $v \in V$.

Output: A harmless set $V' \subseteq V$ such that $|V'|$ is maximized.

If the threshold function is defined by $t(v) = d(v)$ for every $v \in V$ then we add the suffix WITH UNANIMITY to the problem name. The *majority threshold* is $t(v) = \lceil \frac{d(v)}{2} \rceil$ for all $v \in V$.

Parameterized complexity. Here we only give the basic notions on parameterized complexity, for more background the reader is referred to [18, 22, 34]. Parameterized complexity is a framework which provides a new way to express the computational complexity of problems. A problem parameterized by k is called *fixed-parameter tractable* (fpt) if there exists an algorithm, called an fpt algorithm, that solves it in time $f(k) \cdot n^{O(1)}$ (fpt-time) where n is the size of the input. The function f is typically super-polynomial and only depends on k . In other words, the combinatorial explosion is confined into f . The FPT class contains all parameterized problems that are fixed-parameter tractable. The XP class is the set of problems parameterized by k that can be solved in time $n^{g(k)}$ for a given function g .

One of the main tools to design such algorithms is the *kernelization* technique. A kernelization algorithm transforms in polynomial time any instance I of a given problem parameterized by k into an equivalent instance I' of the same problem parameterized by $k' \leq k$ such that the size of I' is bounded by $g(k)$ for some function g . The instance I' is called a *kernel* of size $g(k)$ — if g is a polynomial then I' is a *polynomial kernel*. By applying any algorithm that solves the problem to the reduced instance I' , we directly derive an fpt algorithm (assuming the problem to be decideable). In this paper, the kernel size is expressed in terms of the number of vertices.

Conversely we can prove presumable parameterized intractability of a problem. To this end, we need to introduce the notion of *parameterized reduction*. An fpt-reduction is an algorithm that reduces any instance I of a problem with parameter k to an equivalent instance I' with parameter $k' = g(k)$ in fpt-time for some function g . The basic class of parameterized intractability is W[1] and there is a good reason to believe that W[1]-hard problems — according to the fpt-reduction — are unlikely to be in FPT. In fact there is a hierarchy of classes W[i] with the following inclusions $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \dots \subseteq \text{XP}$. Informally speaking, a prob-

lem in $W[i]$ is considered “harder” than those lying in $W[i - 1]$ where $i > 1$. These classes are defined via the boolean circuit satisfiability problems. More specifically, a parameterized problem belongs to $W[i]$ if every instance (I, k) can be transformed in fpt-time to a boolean circuit C of constant *depth* and *weft* at most i , such that (I, k) is a YES-instance if and only if there is a satisfying truth assignment for C of weight exactly k . The *weft* of a circuit is the maximum number of large gates, *i.e.* gates with a number of inputs not bounded by any constant, on a path from an input to the output. The *depth* is the maximum number of all gates on a path from an input to the output.

Approximation. Given an optimization problem A and an instance I of this problem, we denote by $|I|$ the size of I , by $opt_A(I)$ the optimum value of I and by $val(I, S)$ the value of a feasible solution S of I . For a function $\rho > 1$, an algorithm is a ρ -*approximation* for a maximization problem A if for any instance I of the problem it returns a solution S such that $val(I, S) \geq \frac{opt_A(I)}{\rho(|I|)}$. We say that a maximization problem is *constant approximable* if, for some constant $\rho > 1$, there exists a polynomial-time ρ -approximation for it. A maximization problem has a *polynomial-time approximation scheme* if, for every constant $\varepsilon > 0$, there exists a polynomial-time $(1 + \varepsilon)$ -approximation for it. APX is the class of problems that are constant approximable and PTAS the class of problems that have a polynomial-time approximation scheme. In this paper, we will make use of the following approximation preserving reductions.

Definition 1 (L-reduction [35]) *Let A and B be two optimization problems. Then A is said to be L-reducible to B if there are two constants $\alpha, \beta > 0$ and two polynomial time computable functions f, g such that*

1. f maps an instance I of A into an instance I' of B such that $opt_B(I') \leq \alpha \cdot opt_A(I)$,
2. g maps each solution S' of I' into a solution S of I such that $|val(I, S) - opt_A(I)| \leq \beta \cdot |val(I', S') - opt_B(I')|$.

If a problem A is L-reducible to a problem B then the following holds: If A is APX-hard then B is also APX-hard and if $B \in$ PTAS then $A \in$ PTAS. However, this property is no longer true for a class beyond APX. For instance, let $\varepsilon \in (0, 1)$ be any fixed constant, if a problem A is not $n^{1-\varepsilon}$ -approximable unless $P = NP$ and L-reduces to another problem B then we cannot deduce the same hardness result for B . To get rid of this problem, we need to use the E-reduction.

Definition 2 (E-reduction [30]) *Let A and B be two optimization problems. Then A is said to be E-reducible to B if there exist a constant $\beta > 0$ and two polynomial time computable functions f, g such that*

1. f maps an instance I of A to an instance I' of B such that $opt_A(I)$ and $opt_B(I')$ are related by a polynomial factor, *i.e.* there exists a polynomial p such that $opt_B(I') \leq p(|I|)opt_A(I)$,
2. g maps each solution S' of I' into a solution S of I such that $\varepsilon(I, S) \leq \beta\varepsilon(I', S')$,

where $\varepsilon(I, S) = \max \left\{ \frac{val(I, S)}{opt_A(I)}, \frac{opt_A(I)}{val(I, S)} \right\} - 1$.

For more details about approximation the reader is referred to [2, 28, 40].

Tree decomposition and treewidth. A tree decomposition $\mathcal{T} = (T, \mathcal{H})$ of a graph $G = (V, E)$ consists of a tree $T = (X, F)$ with node set X and edge set F , and a set system \mathcal{H} over V whose members $H_x \in \mathcal{H}$ are labeled with the node $x \in X$, such that the following conditions are met:

1. $\bigcup_{x \in X} H_x = V$.
2. For each $uv \in E$ there is an $x \in X$ with $u, v \in H_x$.
3. For each $v \in V$, the node set $\{x \in X : v \in H_x\}$ induces a subtree of T .

The third condition is equivalent to assuming that if $v \in H_{x'}$ and $v \in H_{x''}$ then $v \in H_x$ holds for every node x of the (unique) $x' - x''$ path in T . The width of a tree decomposition \mathcal{T} is $w(\mathcal{T}) = \max_{x \in X} |H_x| - 1$ and the treewidth of G is defined as $\text{tw}(G) = \min_{\mathcal{T}} w(\mathcal{T})$ where the minimum is taken over all tree decompositions $\mathcal{T} = (T, \mathcal{H})$ of G . The “ -1 ” in the definition of $w(\mathcal{T})$ is included for the convenience that trees have treewidth 1 (rather than 2).

Any tree decomposition $\mathcal{T} = (T, \mathcal{H})$ of a graph can be transformed in linear time into a so-called nice tree decomposition $\mathcal{T}' = (T', \mathcal{H}')$ with $w(\mathcal{T}') = w(\mathcal{T})$, $|\mathcal{H}'| = O(|\mathcal{H}|)$ and with $H_x \neq \emptyset$ for all $H_x \in \mathcal{H}$ where T' is a rooted tree satisfying the following conditions (see [31] for more details):

1. Each node of T' has at most two children.
2. For each node x with two children y, z , we have $H'_y = H'_z = H'_x$ (x is called *join node*) with $H'_x, H'_y, H'_z \in \mathcal{H}'$.
3. If a node x has just one child y , then $H'_x \subset H'_y$ (x is called *forget node*) or $H'_y \subset H'_x$ (x is called *insert node*) and $||H'_x| - |H'_y|| = 1$ with $H'_x, H'_y \in \mathcal{H}'$.

One can see that the subtree T_x of T rooted at node x represents the subgraph G_x induced by precisely those vertices of G which occur in at least one H_y where y runs over the nodes of T_x .

3 Parameterized complexity

In this section, we consider the parameterized complexity of HARMLESS SET. In some reductions we make use of the following gadget: a *forbidden edge* denotes an edge uv where both vertices have threshold one. Attaching a forbidden edge to a vertex w means to create a forbidden edge uv and make w adjacent to u . Notice that none of the three vertices u, v or w can be part of a harmless set. Moreover, we need the following simple but useful data reduction rule.

Data reduction rule 1 *Let (G, t, k) be an instance of HARMLESS SET. If there is a vertex v such that $t(v) > k + 1$ then set the threshold $t(v)$ to $k + 1$ to get a new equivalent instance (G, t', k) .*

To see that the above rule is correct, observe that if $S \subseteq V$ is a harmless set of size at least k for (G, t, k) , then any subset of size k of S is a harmless set for (G, t', k) . The converse is clear.

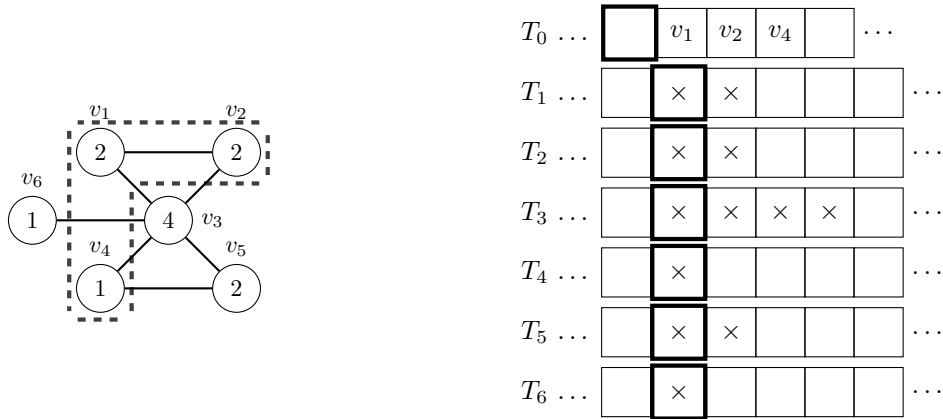


Figure 2: A non-deterministic multi-tape Turing machine (right) accepting an instance of HARMLESS SET where $k = 3$ (left). The guessed solution is the set $\{v_1, v_2, v_4\}$.

We now show that HARMLESS SET belongs to $W[2]$ using the *Turing way*, that is, we reduce HARMLESS SET to the SHORT MULTI-TAPE NONDETERMINISTIC TURING MACHINE problem that is proved to belong to $W[2]$ in [11] and is defined as follows. Given a multi-tape nondeterministic Turing machine M , a word x on the input alphabet of M , and an integer k , determine if there is a computation of M on input x that reaches a final accepting state in at most k steps. The parameter is k .

Proposition 3 HARMLESS SET is in $W[2]$.

Proof: We construct an fpt-reduction from HARMLESS SET to SHORT MULTI-TAPE NONDETERMINISTIC TURING MACHINE as follows. Let (G, t, k) be an instance of HARMLESS SET with $G = (V, E)$ and $V = \{v_1, \dots, v_n\}$. First, exhaustively apply Data reduction rule 1 to obtain a new equivalent instance (G, t', k) . Then construct the following Turing machine M from (G, t', k) (see Figure 2). We create $n + 1$ tapes denoted by $T_0, T_{v_1}, \dots, T_{v_n}$. The tape alphabet is $V \cup \{\times\}$ plus the blank symbol \square . Initially, every tape is filled with \square . The transition function is defined hereafter. First, M non-deterministically chooses k vertices and writes them on tape T_0 , that is, if M picks a vertex $v \in V$ then it writes symbol v on T_0 and moves T_0 's head one step to the right. The previous procedure is done in k steps. Next, for each $i = 1, \dots, k + 1$, the Turing machine writes a symbol \times on each tape T_{v_j} and moves T_{v_j} 's head one step to the right if v_j has a threshold greater or equal to i . Recall that no vertex has a threshold greater than $k + 1$ due to Data reduction rule 1. During the third phase, M checks whether the selected set is a harmless set as follows. First, the machine moves all heads one step to the left. If T_0 's head reads symbol v then for every $u \in N(v)$, we simply move T_u 's head one step to the left. We repeat the previous procedure until T_0 's head reads a blank symbol. If all the other tapes read a \times symbol then M goes in an accepting state; otherwise it goes to a rejecting state. This checking phase can be done in at most $k + 1$ steps. Finally, the input word x is empty and $k' = 3k + 2$. It is not hard to see that (G, t, k) is a YES-instance if and only if there is a computation of M that accepts x in at most k' steps. \square

Now in order to prove the $W[2]$ -hardness of HARMLESS SET, we construct a simple fpt-reduction from RED/BLUE DOMINATING SET defined as follows. Given a bipartite graph $G = (R \cup B, E)$ and a positive integer k , determine if there exists a set $R' \subseteq R$ of cardinality k such

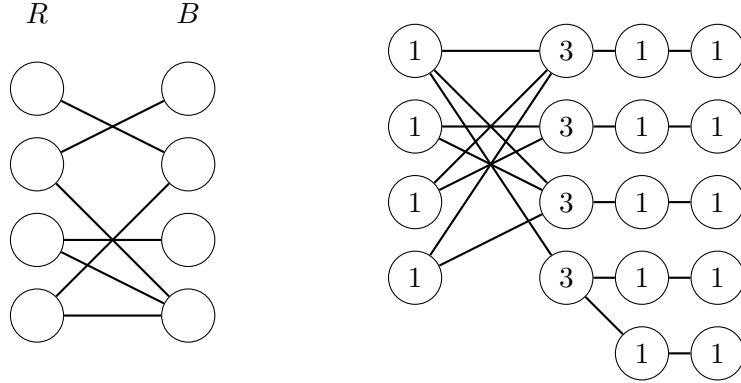


Figure 3: Illustration of the construction of G' (right) from the graph G (left) after carrying out the modifications of Theorem 4 with $k = 3$.

that every vertex in B has at least one neighbor in R' . The parameter is k . The RED/BLUE DOMINATING SET problem is equivalent to HITTING SET and, thus, W[2]-hard [18].

Theorem 4 HARMLESS SET is W[2]-complete even on bipartite graphs.

Proof: Membership follows from Proposition 3. Now, let us show the W[2]-hardness. Let $(G = (R \cup B, E), k)$ be an instance of RED/BLUE DOMINATING SET, we construct an instance $(G' = (V', E'), t, k)$ of HARMLESS SET as follows. We consider the complement \bar{G} of the graph G , that is two vertices $u \in R$ and $v \in B$ are adjacent in \bar{G} if and only if they are not adjacent in G . Moreover, the sets R and B remain independent sets. Graph G' is obtained from \bar{G} by attaching $\max\{k - d_{\bar{G}}(v), 1\}$ forbidden edges to each vertex $v \in B$. Finally, set $t(v) = k$ for every vertex $v \in B$ and $t(v) = 1$ for every vertex $v \in R$. Adding several forbidden edges to the vertices of B ensures that the threshold of these vertices is less than or equal to their degree as required (see Figure 3).

Assume that (G, k) has a solution $R' \subseteq R$ of size k . One can see that R' is also a solution for (G', t, k) since no vertex in B is adjacent to all vertices in R' . Conversely, suppose that there is a harmless set $S \subseteq V'$ of size k in G' . Since S is harmless, S cannot contain any vertex from B because of the forbidden edges, and thus S is entirely contained in R . Moreover, every vertex v in B is adjacent in G' to at most $t(v) - 1 = k - 1$ vertices in S . Hence, every vertex in B is adjacent in G to at least one vertex in S . Therefore, S is a solution of size k for (G, k) . \square

In the next two theorems, we show that HARMLESS SET goes one level down in the W-hierarchy when all thresholds are bounded by a constant.

Proposition 5 HARMLESS SET is in W[1] if all thresholds are bounded by a constant.

Proof: Let $(G = (V, E), t, k)$ be an instance of HARMLESS SET where $t(v) \leq c$ for every $v \in V$ and some constant $c > 0$. We construct in $O(n^c)$ -time, where n is the number of vertices of G , a boolean circuit C of depth 3 and weft 1 as follows. We identify the inputs of the circuit with the vertices of G . Connect a \neg -gate to every input. For all $v \in V$ and all subsets $S' \subseteq N(v)$ of size $t(v)$, add a \vee -gate connected to the \neg -gates of inputs in S' . Finally, add a large \wedge -gate connected to every \vee -gate. It is not hard to see that G admits a harmless set of size k if and only if there is a weight- k assignment that satisfies C . \square

We establish the W[1]-hardness of HARMLESS SET by an fpt-reduction from the CLIQUE problem [17] defined as follows. Given an undirected graph $G = (V, E)$ and a positive integer k , determine if there is a clique $C \subseteq V$ of size at least k . The parameter is k .

Theorem 6 HARMLESS SET *is*

1. W[1]-hard even on bipartite graphs with majority thresholds.
2. W[1]-complete even on split graphs, i.e. graphs whose vertices can be partitioned into a clique and an independent set, with thresholds $t(v) = 2$ for every vertex v .

Proof:

(1) Given $(G = (V, E), k)$ an instance of CLIQUE, we construct an instance $(G' = (V', E'), t, k)$ of HARMLESS SET as follows. The set V' is obtained from V by adding for each non-edge $uv \notin E$, an edge-vertex e_{uv} to V' and edges ue_{uv} and $e_{uv}v$ to E' . Remove every edge in E . Attach a forbidden edge $p_{uv}q_{uv}$ to each edge-vertex e_{uv} . Finally, set $t(v) = \lceil \frac{d(v)}{2} \rceil$ for all $v \in V'$. Observe that the vertices in a forbidden edge have majority thresholds (see Figure 4).

Let $C \subseteq V$ be a clique of size at least k . Then C is clearly a harmless set in G' since no edge-vertex has more than one neighbor in C . Conversely, let $C' \subseteq V'$ be a harmless set in G' . Because of the forbidden edges, C' cannot contain an edge-vertex e_{uv} and p_{uv}, q_{uv} and thus $C' \subseteq V$. Moreover, since $t(e_{uv}) = 2$, the set C cannot contain u and v such that $uv \notin E$ and thus C' is a clique of size at least k in G .

(2) Membership follows from Proposition 5. We now prove the W[1]-hardness. Let $(G = (V, E), k)$ be an instance of CLIQUE, we construct an instance $(G' = (V', E'), t, k)$ of HARMLESS SET as follows. As previously, for each non-edge $uv \notin E$, add an edge-vertex e_{uv} and the edges ue_{uv} and $e_{uv}v$. Add edges to make the set of all edge-vertices a clique. Remove every edge in E . Finally, set $t(v) = 2$ for all $v \in V'$. Without loss of generality we may assume that $k \geq 2$ and every vertex in V has minimum degree two (see Figure 4).

Let $C \subseteq V$ be a clique of size at least k . One can easily verify that C is a harmless set in G' . Conversely, suppose that there is a harmless set $C' \subseteq V'$ of size k . Notice that $C' \subseteq V$ since otherwise we would not have been able to take more than one vertex in G' . Indeed, if there are two vertices $u, v \in C'$ with $v \in V' \setminus V$ then there is always a vertex $w \in V' \setminus V - \{u, v\}$ adjacent to both v and u . Thus, C' is entirely contained in V . Now, it is not hard to see that C' is a clique of size at least k in G . \square

It is interesting to note that the ratio between the number n_u of vertices with unbounded threshold over the total number of vertices in G' in the proof of Theorem 4 can be made arbitrarily small by adding many forbidden edges. This implies a sharp dichotomy between the W[2]- and W[1]-completeness of HARMLESS SET in the following sense. Let r denote the ratio $\frac{n_u}{n} \geq 0$ where n is the order of the input graph. For any fixed $\varepsilon > 0$, the HARMLESS SET problem is W[2]-complete even when $r < \varepsilon$ and W[1]-complete when $r = 0$.

Unanimity thresholds. Now we consider the HARMLESS SET WITH UNANIMITY problem. First, we start with the following easy observation. In the case of unanimity thresholds, any harmless set is the complement of a total dominating set. Recall that a total dominating set S is a set of vertices such that every vertex in the input graph has at least one neighbor in S . Moreover, we have the following theorem.

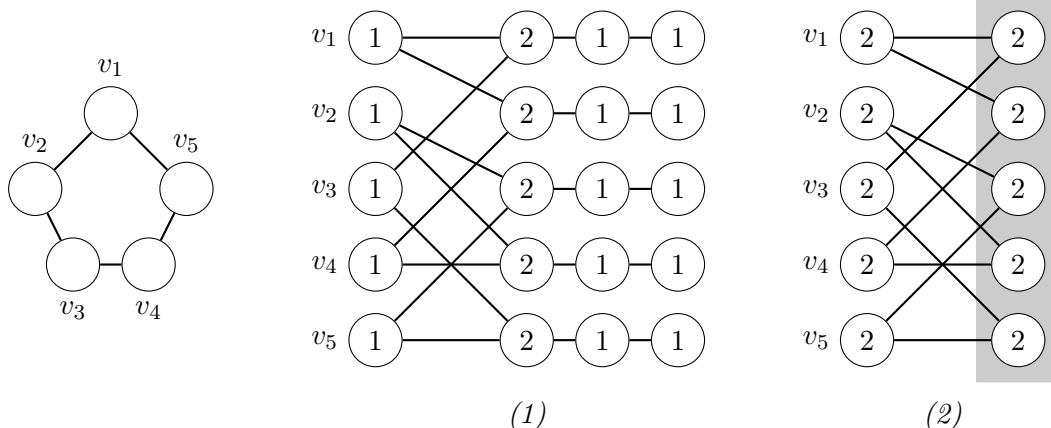


Figure 4: Illustration of the reductions (1) and (2) from CLIQUE as described in Theorem 6. The vertices in the grey box form a clique.

Theorem 7 (Cockayne et al. [15]) *If G is a connected graph of order at least 3 then there is a total dominating set of size at most $2n/3$.*

Now we can prove the following

Theorem 8 HARMLESS SET WITH UNANIMITY *admits a kernel with $3k$ vertices.*

Proof: Let $(G = (V, E), k)$ be an instance of HARMLESS SET WITH UNANIMITY. The aim of the proof is to apply Theorem 7 on G . For this to work, we need to get rid of connected components of size 1 and 2 using the following two reduction rules.

1. If there is an isolated vertex v (i.e. $d(v) = 0$) then delete v from G and reduce k by one.
2. If there is an isolated edge $uv \in E$ (i.e. $d(u) = d(v) = 1$) then delete u and v from G .

The correctness of the above rules follows from the fact that every isolated vertex is included in any maximal harmless set while both endpoints of an isolated edge must be excluded.

Let (G', k') be the instance obtained after exhaustively applying the above rules. Let n' be the order of G' . From Theorem 7, we know that there exists a harmless set in G' of size least $n'/3$. Hence, if $k' \leq n'/3$ then return a trivial YES-instance. If $k' > n'/3$ then (G', k') is a kernel of size at most $3k'$. \square

Observe that the parameter k might be “large” in the previous kernel. This suggests to look for other parameterizations. One possibility is to decide the existence of solutions of size at least $\lceil \frac{n}{3} \rceil + k$. Another one is to decide the existence of solutions of size at least $n - k$. We study in the following this last problem and leave as an open question the first one.

Parametric dual. Notice that $(n - k)$ -HARMLESS SET with unanimity thresholds is exactly the TOTAL DOMINATING SET problem which is known to be W[2]-hard with respect to the solution size [23]. Nonetheless the parameterized tractability of the problem for other threshold functions is open. In what follows, we show that $(n - k)$ -HARMLESS SET is in FPT with respect to the parameter k for a large family of threshold functions including majority and constant thresholds. Toward this goal, we provide a kernelization through the following data reduction rule.

Data reduction rule 2 Let (G, t, k) be an instance of $(n - k)$ -HARMLESS SET. If there is a vertex v such that $d(v) \geq k + t(v) - 1$ then remove v and decrease by one the threshold of every vertex in $N(v)$ to get a new equivalent instance (G', t', k) .

Regarding the correctness of the above rule, let S be a harmless set of size at least $n - k$. Notice that if there is a vertex v with $d(v) \geq k + t(v) - 1$ then v must be in S since otherwise it will have at most $k - 1$ neighbors outside S and then at least $t(v)$ neighbors in S .

We can now state the main result.

Theorem 9 $(n - k)$ -HARMLESS SET admits a kernel with $O(k^2)$ vertices if for every vertex v in the input graph $t(v) = \lceil \alpha_v d(v)^{\beta_v} + \gamma_v \rceil$ for any fixed constants $\alpha_v, \beta_v \in [0, 1], \alpha_v \beta_v \neq 1$, and $\gamma_v \in \mathbb{Q}$.

Proof: Let (G, t, k) be an instance of $(n - k)$ -HARMLESS SET. Exhaustively apply Data reduction rule 2 to get (G', t', k) . Assume that there exists a solution $S \subseteq V$ of size at least $n - k$. Because of Data reduction rule 2, we have

$$d(v) < k + t(v) - 1 = k + \lceil \alpha_v d(v)^{\beta_v} + \gamma_v \rceil - 1 \leq k + \alpha_v d(v)^{\beta_v} + \gamma_v \quad (1)$$

We claim that $d(v) \leq \theta_v(k)$ for all $v \in V'$ where

$$\theta_v(k) = \begin{cases} k + \alpha_v + \gamma_v & \text{if } \beta_v = 0 \\ \frac{k + \gamma_v}{1 - \alpha_v} & \text{if } \beta_v = 1 \\ \frac{k + \gamma_v}{1 - \beta_v} + (1/\beta_v)^{\frac{1}{1 - \beta_v}} & \text{otherwise} \end{cases}$$

The first two cases are straightforward. Suppose now that $\beta_v \in (0, 1)$. First, it is not hard to show that the following holds: $x^\varepsilon \leq \varepsilon x$ if and only if $x \geq (1/\varepsilon)^{\frac{1}{1 - \varepsilon}}$ for any $x \geq 1$ and $\varepsilon \in (0, 1)$. Hence, if $d(v) \geq (1/\beta_v)^{\frac{1}{1 - \beta_v}}$ then, together with inequality (1), we obtain $d(v) \leq k + \alpha_v \beta_v d(v) + \gamma_v$ and thus $d(v) \leq \frac{k + \gamma_v}{1 - \alpha_v \beta_v} \leq \theta_v(k)$. Otherwise $d(v) < (1/\beta_v)^{\frac{1}{1 - \beta_v}} \leq \theta_v(k)$.

Since every vertex from S has at least one neighbor in $V' - S$ then $|S|$ has at most $|V' - S| d_{\max} \leq k \theta_{\max}(k)$ vertices where $\theta_{\max}(k) = \max_{v \in V'} \theta_v(k)$ and d_{\max} is the maximum degree of vertices in $V' - S$.

The kernelization procedure is then defined as follows. From an instance (G, t, k) of $(n - k)$ -HARMLESS SET, exhaustively apply Data reduction rule 2 to get an instance (G', t', k) . If $|V'| > k \theta_{\max}(k) + k$ then return a trivial NO-instance. Otherwise, return the reduced instance (G', t', k) . \square

4 Algorithms for trees and tree-like graphs

In this section we establish a $t_{\max}^{O(\omega)} \cdot n$ -time algorithm for MAX HARMLESS SET and a $k^{O(\omega)} \cdot n$ -time algorithm for HARMLESS SET where t_{\max} is the maximum threshold and ω the width of a given tree decomposition of the input graph. We first describe an $O(\log(t_{\max}) \cdot n)$ -time algorithm for trees. Besides to be more efficient in this case, this algorithm introduces the underlying ideas used later for the algorithm on general graphs.

Proposition 10 MAX HARMLESS SET is solvable in $O(\log(t_{\max}) \cdot n)$ time on trees.

Proof: Let $(T = (V, E), t)$ be an instance of MAX HARMLESS SET where T is a tree rooted at r . We describe a dynamic programming algorithm as follows. We denote by T_v the subtree of T rooted at v . Moreover, we denote by $C(v)$ the set of children of v and $p(v)$ the parent of v .

For each $v \in V$ and each $b \in \{0, 1\}$, we define $I_v[b]$ (resp. $E_v[b]$) as the optimal solution for the subtree T_v with the additional constraints that $t(v)$ is decreased by b and v is included (resp. excluded). When no harmless set satisfying the constraints exists, we set the value \perp .

As a preliminary step, we set $F(p(v)) = \perp$ whenever $t(v) = 1$ for every vertex $v \neq r$. Thus a vertex v such that $F(v) = \perp$ cannot be part of any solution since it has a threshold one neighbor.

For every leaf v of T (recall that $t(v) = 1$ in this case), set $I_v[0] = \{v\}$, $E_v[0] = \emptyset$, and $I_v[1] = E_v[1] = \perp$. For each non-leaf vertex v perform the following steps

1. For each child $c \in C(v)$ and each $b \in \{0, 1\}$, compute the *magnitude* value $m(c, b) = |I_c[b]| - |E_c[b]|$.
2. For each $b_1 \in \{0, 1\}$ and $b_2 \in \{0, 1\}$, partition the set $C(v)$ into two sets $C_1(v, b_1, b_2)$ and $C_2(v, b_1, b_2)$ such that $C_1(v, b_1, b_2)$ contains the $t(v) - b_1 - 1$ vertices with the largest positive magnitude value $m(c, b_2)$, that is every pair $c_1, c_2 \in C_1(v, b_1, b_2) \times C_2(v, b_1, b_2)$ verifies $m(c_1, b_2) \geq m(c_2, b_2)$ and $m(c_1, b_2) > 0$. If $t(v) - b_1 - 1 \leq 0$ then set $C_1(v, b_1, b_2) = \emptyset$.
3. For each $b \in \{0, 1\}$, update $I_v[b]$ and $E_v[b]$ as follows

$$I_v[b] = \begin{cases} \perp & \text{if } F(v) = \perp \\ \perp & \text{if } t(v) = 1 \text{ and } b = 1 \\ \{v\} \cup \bigcup_{c \in C_1(v, b, 1)} I_c[1] \cup \bigcup_{c \in C_2(v, b, 1)} E_c[1] & \text{otherwise} \end{cases} \quad (1)$$

$$E_v[b] = \begin{cases} \perp & \text{if } t(v) = 1 \text{ and } b = 1 \\ \bigcup_{c \in C_1(v, b, 0)} I_c[0] \cup \bigcup_{c \in C_2(v, b, 0)} E_c[0] & \text{otherwise} \end{cases} \quad (2)$$

In the above equations, we adopt the convention that $Q \cup \perp = \perp$ for any set Q . To get the optimal solution for the tree T , return the largest solution between $I_r[0]$ and $E_r[0]$.

As to the correctness, notice that when we make a decision for a vertex v , we do not know the decision about its parent. We then have to deal with two cases: one where the parent is in the solution and the other one when it is not. The first case can be handled by computing an optimal solution with the threshold of v set to $t(v) - 1$. For the second case, we compute another optimal solution without modifying v 's threshold. Notice that in each case, the optimal solution for T_v takes either v (Eq. 1) or not (Eq. 2). Therefore the subtree T_v is associated with four optimal solutions $I_v[0]$, $E_v[0]$, $I_v[1]$, and $E_v[1]$.

We now prove equation (1) (equation (2) is proved using similar arguments). In the first case ($F(v) = \perp$) there exists a child c of v such that $t(c) = 1$ and thus v cannot be part of any harmless set thus $I_v[b] = \perp$. In the second case, we set $I_v[b] = \perp$ since the parent of v is

included in the solution ($b = 1$) while $t(v) = 1$. Consider now the third case. For simplicity, let us assume that the parent of v is taken in the solution and then we want to update $I_v[1]$. Observe that we cannot add more than $t(v) - 2$ children of v in the solution and that for each child $c \in C(v)$ we need to determine what is the best to include between $I_c[1]$ and $E_c[1]$. To this end, we simply compute the so-called magnitude value of each child c of v which corresponds to the gain we obtain by choosing to include $I_c[1]$ instead of $E_c[1]$. Now, let us associate a binary variable x_c to each child c where $x_c = 1$ if $I_c[1]$ is chosen and $x_c = 0$ if $E_c[1]$ is taken. The optimal solution $I_v[1]$ is then equal to

$$\{v\} \cup \bigcup_{c:x_c^*=1} I_c[1] \cup \bigcup_{c:x_c^*=0} E_c[1]$$

where $\{x_c^*\}_{c \in C(v)}$ is the solution that maximizes $\sum_{c \in C(v)} |I_c[1]|x_c + |E_c[1]|(1 - x_c)$ (or, equivalently, maximizes $\sum_{c \in C(v)} m(c, 1)x_c$) subject to $\sum_{c \in C(v)} x_c \leq t(v) - 2$. Now it should be clear that $C_1(v, 1, 1) = \{c \in C(v) : x_c^* = 1\}$. Therefore the equation correctly updates $I_v[1]$.

As to the running time, observe that the preliminary step as well as the initialization of the leaf's tables can be done in $O(n)$ time. Furthermore, the number of steps performed in each non-leaf vertex is $O(\log(t_{\max}) \cdot |C(v)|)$. Indeed, partitioning the set $C(v)$ (Step 2) requires $O(\log(t_{\max}) \cdot |C(v)|)$ time and Step 1 and Step 3 both uses $O(|C(v)|)$ time. Overall, the running time is $O(\sum_{v \in V} \log(t_{\max}) \cdot |C(v)|) = O(\log(t_{\max}) \cdot n)$. This completes the proof. \square

Now, we present the algorithm for solving MAX HARMLESS SET on general graphs.

Theorem 11 *Given a tree decomposition of width ω of a graph G , a maximum harmless set can be computed in time $t_{\max}^{O(\omega)} \cdot n$ where t_{\max} is the maximum threshold.*

Proof: Let $(G = (V, E), t)$ be an instance of MAX HARMLESS SET. Assume that we are given a nice tree decomposition $\mathcal{T} = (T = (X, F), \mathcal{H})$ of G of width at most ω . Let T_x be the subtree of T rooted at some node $x \in X$. We denote by $G_x = (V_x, E_x)$ the subgraph induced by the vertices from $\bigcup_{y \in T_x} H_y$. We describe a dynamic programming algorithm to solve (G, t) using \mathcal{T} .

Description. The general idea of the algorithm is as follows. For each node $x \in X$, we store a set of optimal solutions for the subgraph G_x in a table denoted by A_x . These tables are updated using a bottom-up procedure that starts from the leaves and ends at the root of T . More precisely, we use a two-entry table $A_x[\mathbf{t}, \mathbf{c}]$ where $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$, $\mathbf{c} \in \{0, 1\}^{H_x}$, and $A_x[\mathbf{t}, \mathbf{c}]$ corresponds to a maximum harmless set in G_x whose intersection with H_x is exactly $\{v \in H_x : \mathbf{c}(v) = 1\}$ and by imposing the threshold $t(v) = \mathbf{t}(v)$ for every $v \in H_x$. We set $A_x[\mathbf{t}, \mathbf{c}] = \perp$, if no such harmless set is possible. During the computation, we may interrogate the table A_x with $\mathbf{t}(v) \leq 0$ for some $v \in H_x$. In such case, the table simply returns the value \perp .

Consider the updating step occurring in a join node. Let $x \in X$ be a join node with children y and z such that $H_x = H_y = H_z$. The nodes y and z have their respective tables A_y and A_z already computed by dynamic programming and we want to compute the table A_x . Let $B = N_G(H_x) \setminus V_x$. Notice that, when computing A_x , we do not know what vertices in B will be in the maximum harmless set. Thus, one has to take into consideration that any subset $S \subseteq B$ might be in the optimal solution. Hence, we have to compute a

maximum harmless set in G_x for each subset $S \subseteq B$ considering S as part of a harmless set. This can be done by computing a maximum harmless set in G_x for every possible thresholds $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$. To do this, we first need to solve the following two problems. Consider the optimal solution S_y (resp. S_z) in G_y (resp. G_z) by imposing the thresholds \mathbf{t} to H_y (resp. H_z) for some $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$. We cannot directly make the union of S_y and S_z to get the optimal solution S_x of G_x under the restriction that vertices of H_x have thresholds set to \mathbf{t} . Indeed, consider a vertex $u \in H_x$. It may happen that u has less than $\mathbf{t}(u)$ neighbors in S_y and S_z but more than $\mathbf{t}(u)$ in $S_y \cup S_z$. So as to overcome this situation, we have to consider the union of a pair of optimal solutions S_y, S_z for each possible thresholds value $\mathbf{t}_1 \in \{1, \dots, t_{\max}\}^{H_y}$ and $\mathbf{t}_2 \in \{1, \dots, t_{\max}\}^{H_z}$ of the vertices in H_y and H_z , respectively, with $\mathbf{t}_1 + \mathbf{t}_2 = \mathbf{t}$ ($\mathbf{t}_1 + \mathbf{t}_2$ is the function defined as $(\mathbf{t}_1 + \mathbf{t}_2)(u) = \mathbf{t}_1(u) + \mathbf{t}_2(u)$ for every $u \in H_x$).

The other problem is whenever we make the previous union, we do not take into consideration that the sets H_y and H_z are equal. The consequence is that a vertex v in G_y might have a number of neighbors in $H_y \cap (S_y \cup S_z)$ that sums over its threshold. We solve this issue using the function \mathbf{c} . According to the definition of $A_x[\mathbf{t}, \mathbf{c}]$, this function ensures that the same vertices in both H_y and H_z are in the solution. Observe that a vertex v in H_x taken in the solution ($\mathbf{c}(v) = 1$) may be adjacent to some vertex u in H_x and thus affects $\mathbf{t}(u)$. Since we consider H_y and H_z separately, we count the vertex v once for $\mathbf{t}_1(u)$ and a second time for $\mathbf{t}_2(u)$. This problem can be overcome by simply increasing the thresholds of u in both H_x and H_y so as to balance this overcounting.

This completes the description of the algorithm, we now give the formal details.

Algorithm. We denote by $\mathbf{f} \oplus (a, b)$ the extended function defined as $(\mathbf{f} \oplus (a, b))(x) = \mathbf{f}(x)$ if $x \neq a$ and $(\mathbf{f} \oplus (a, b))(a) = b$. Similarly to the proof of Proposition 10, we adopt the convention that $Q \cup \perp = \perp$ for any set Q .

Initialization step. We initialize all the tables A_x where x is a leaf of T as follows. For each leaf x of T , $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$ and $\mathbf{c} \in \{0, 1\}^{H_x}$. Let $S = \{v \in H_x : \mathbf{c}(v) = 1\}$

$$A_x[\mathbf{t}, \mathbf{c}] = \begin{cases} S & \text{if } S \text{ is a harmless set for } G_x \text{ according to } \mathbf{t} \\ \perp & \text{otherwise} \end{cases}$$

Updating step. Starting from the leaves, we apply the following rules to each node $x \in X$ we visit until we reach the root.

Case 1 (insert node). Suppose that x is an insert node with child y such that $H_x = H_y \cup \{u\}$. Following the above discussion, we update the table A_x as follows. For all $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_y}$, $\mathbf{c} \in \{0, 1\}^{H_y}$ and $i = 1, \dots, t_{\max}$

$$A_x[\mathbf{t} \oplus (u, i), \mathbf{c} \oplus (u, 0)] = \begin{cases} A_y[\mathbf{t}, \mathbf{c}] & \text{if } A_y[\mathbf{t}, \mathbf{c}] \text{ is a harmless set in } G_x \text{ with } t(u) = i. \\ \perp & \text{otherwise} \end{cases}$$

$$A_x[\mathbf{t} \oplus (u, i), \mathbf{c} \oplus (u, 1)] = \begin{cases} A_y[\mathbf{t}', \mathbf{c}] \cup \{u\} & \text{if } A_y[\mathbf{t}', \mathbf{c}] \cup \{u\} \text{ is a harmless set in } G_x \\ & \text{with } t(u) = i \\ \perp & \text{otherwise} \end{cases}$$

where $\mathbf{t}'(v) = \mathbf{t}(v) - 1$ if $v \in N(u)$ and $\mathbf{t}'(v) = \mathbf{t}(v)$ otherwise.

Case 2 (forget node). Suppose that x is a forget node with child y such that $H_x = H_y - \{u\}$. Let $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_y}$. Notice that vertex u has its neighbors entirely contained

in G_x . Hence, the maximum harmless set for G_x where $t(v) = \mathbf{t}(v)$ for all $v \in H_x$ is exactly the maximum harmless set for G_y where $t(v) = \mathbf{t}(v)$ for all $v \in H_y$ and such that u has threshold $t(u)$. Formally, we update the table A_x as follows. For all $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$ and $\mathbf{c} \in \{0, 1\}^{H_x}$

$$A_x[\mathbf{t}, \mathbf{c}] = \max_{i \in \{0, 1\}} \{A_y[\mathbf{t} \oplus (u, t(u)), \mathbf{c} \oplus (u, i)]\}$$

Case 3 (join node). Suppose that x is a join node with children y and z such that $H_x = H_y = H_z$. According to the above discussion, we update the table A_x as follows. For all $\mathbf{t} \in \{1, \dots, t_{\max}\}^{H_x}$ and $\mathbf{c} \in \{0, 1\}^{H_x}$, perform the following steps

- Set $\mathbf{e}(u) = |\{v \in H_x : u \in N(v) \text{ and } \mathbf{c}(v) = 1\}|$ for all $u \in H_x$.
- $(\mathbf{t}_1^*, \mathbf{t}_2^*) = \underset{\substack{\mathbf{t}_1, \mathbf{t}_2 \in \{1, \dots, t_{\max}\}^{H_x} \\ : \mathbf{t}_1 + \mathbf{t}_2 = \mathbf{t} - \mathbf{e}}}{\arg \max} |A_y[\mathbf{t}_1 + \mathbf{e}, \mathbf{c}] \cup A_z[\mathbf{t}_2 + \mathbf{e}, \mathbf{c}]|$
- $A_x[\mathbf{t}, \mathbf{c}] = A_y[\mathbf{t}_1^* + \mathbf{e}, \mathbf{c}] \cup A_z[\mathbf{t}_2^* + \mathbf{e}, \mathbf{c}]$

Final step. The optimal solution is then $\arg \max_{\mathbf{c} \in \{0, 1\}^{H_r}} |A_r[\mathbf{t}_r, \mathbf{c}]|$ where r is the root of T and $\mathbf{t}_r(v) = t(v)$ for all $v \in H_r$. As to the running time observe that enumerating all possible functions \mathbf{t} and \mathbf{c} takes $t_{\max}^{O(\omega)}$ time and that updating the table can be done in linear time. This completes the proof. \square

Now we show that HARMLESS SET is fixed-parameter tractable with respect to the combined parameter k and treewidth of the input graph. For that purpose consider an instance $I = (G, t, k)$ of HARMLESS SET. After applying Data reduction rule 1 exhaustively on I we get in polynomial-time a new equivalent instance $I' = (G, t', k)$ such that the threshold value of every vertex is bounded by $k + 1$ and thus $t_{\max} = k + 1$. We can apply the algorithm of Theorem 11 on (G, t') and then decide whether the given solution is greater or equal k . This gives us the following result.

Proposition 12 *Given a tree decomposition of width ω of the input graph HARMLESS SET is solvable in time $k^{O(\omega)} \cdot n$.*

Since a tree decomposition of width $\text{tw}(G)$ of a graph G can be found in fpt-time with respect to $\text{tw}(G)$ [8], it follows that HARMLESS SET is in FPT with respect to the combined parameter k and the treewidth of the input graph.

Notice that the above results are of purely theoretical interest as the current running time makes the treewidth algorithm not practical. Using techniques like fast subset convolution may help in speeding up the algorithm [39].

5 Approximability

We first observe that MAX HARMLESS SET is inapproximable even for majority and small constant thresholds. In order to prove this result, we consider the MAX CLIQUE problem: given a graph $G = (V, E)$, find a clique $C \subseteq V$ of maximum size.

Theorem 13 *If $NP \neq ZPP$, MAX HARMLESS SET is not approximable within $n^{\frac{1}{2}-\varepsilon}$ for any $\varepsilon > 0$ even*

1. *For bipartite graphs with majority thresholds.*
2. *For split graphs with thresholds $t(v) = 2$ for every vertex v .*

Proof: We construct an E -reduction (see Definition 2) from MAX CLIQUE. Let G be an instance of MAX CLIQUE. Consider the constructed instance $I' = (G', t)$ from G as it is defined in Theorem 6. Let C be a harmless set in G' . From the proof of Theorem 6, we know that C is a clique in G . Thus, it is not hard to see that $opt(I') = opt(G)$ and $\varepsilon(G, C) = \varepsilon(I', C)$. Let n and n' be the orders of G and G' , respectively. Since MAX CLIQUE is not approximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ unless $NP = ZPP$ [24] and $n' = O(n^2)$, the result follows. \square

We now prove the APX-completeness of MAX HARMLESS SET WITH UNANIMITY.

Proposition 14 *MAX HARMLESS SET WITH UNANIMITY is 3-approximable in $O(\log(\Delta) \cdot n)$ -time where Δ is the maximum degree of the input graph.*

Proof: Let $G = (V, E)$ be an instance of MAX HARMLESS SET WITH UNANIMITY. We denote by V_1 the set of isolated vertices and by V_2 the set of vertices corresponding to endpoints of isolated edges in G . Let $V_{\geq 3} = V \setminus (V_1 \cup V_2)$. The algorithm consists of the following two steps:

1. Compute a spanning forest T of G .
2. Compute an optimal solution S of T using Proposition 10 with unanimity thresholds.

Observe that any feasible solution S for T is also a solution for G . Indeed, if a vertex v in T is such that $N_T(v) \not\subseteq S$ then we have $N_G(v) \not\subseteq S$. Observe also that no vertices in V_2 can be part of a solution and any maximal solution contains V_1 . Hence, using Theorem 7, we know that $|S| \geq |V_1| + |V_{\geq 3}|/3$. Moreover, $opt(G) \leq |V_1| + |V_{\geq 3}|$. It follows that $|S| \geq opt(G)/3$. \square

Theorem 15 *MAX HARMLESS SET WITH UNANIMITY is APX-complete even on bipartite graphs.*

Proof: Membership follows from Proposition 14. In order to prove the APX-hardness we provide an L-reduction (see Definition 1) from MAX E2SAT-3 proved APX-hard in [7] and defined as follows: given a CNF formula ϕ with n variables and m clauses, in which every clause contains exactly two literals and every variable appears in exactly three clauses, determine an assignment to the variables satisfying a maximum number of clauses. Notice that $m = 3n/2$.

Given a formula ϕ of MAX E2SAT-3, we construct an instance $I = (G = (V, E))$ of MAX HARMLESS SET WITH UNANIMITY as follows (see Figure 5).

- For every variable x_i , we construct the complete bipartite graph $K_{3,3}(x_i) = (V^-(x_i), V^+(x_i))$ in which every edge uv is replaced by an edge-vertex e_{uv} and two edges ue_{uv} and $e_{uv}v$. We denote by $E(x_i)$ this set of edge-vertices. The vertices in $V^+(x_i)$ (resp. $V^-(x_i)$) represent the positive (resp. negative) literals of x_i .

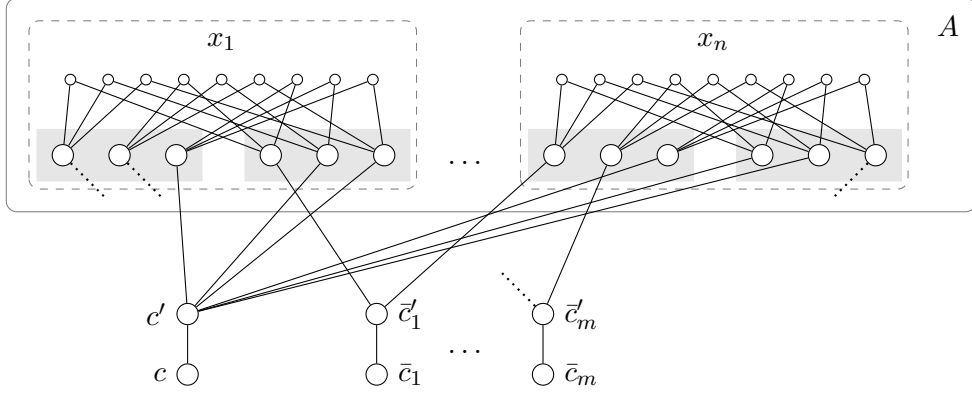


Figure 5: The construction of G . Each subgraph in a dashed box corresponds to a gadget $K_{3,3}(x_i)$ of a variable x_i . A grey box represents either $V^+(x_i)$ or $V^-(x_i)$.

We denote by A the set of all vertices added so far.

- For every clause c_j in ϕ add two adjacent clause-vertices \bar{c}_j and \bar{c}'_j .
- For every variable x_i , if x_i appears positively (resp. negatively) in a clause c_j then choose a vertex v of $V^-(x_i)$ (resp. $V^+(x_i)$) not adjacent to a clause-vertex and add the edge $\bar{c}'_j v$. Thus, vertex \bar{c}'_j represents the complement of the clause c_j in ϕ .
- Finally, add two adjacent vertices c and c' . For every vertex $v \in V^-(x_i) \cup V^+(x_i)$, if v is not adjacent to a clause-vertex then add the edge vc' .

This completes the construction. From the proof of [7], we may assume that each variable of ϕ appears positively and negatively. Thus c' is adjacent to at least one vertex of $V^-(x_i)$ and one vertex of $V^+(x_i)$, *i.e.*

$$N(c') \cap V^-(x_i) \neq \emptyset \quad (1)$$

and

$$N(c') \cap V^+(x_i) \neq \emptyset \quad (2)$$

for each $i = 1, \dots, n$.

Observe that the optimal value in I is bounded by the number of vertices of G and thus, $opt(I) \leq 15n + 2m + 2 \leq 16opt(\phi) + 2$ which implies

$$opt(I) \leq 18opt(\phi)$$

since $opt(\phi) \geq 3/4m$ and $opt(\phi) \geq 1$.

Moreover, let x^* be an optimal assignment for ϕ and let

$$S = \bigcup_{i:x_i^*=1} V^+(x_i) \cup \bigcup_{i:x_i^*=0} V^-(x_i) \cup \bigcup_{i=1}^n E(x_i) \cup \{\bar{c}_j : c_j \text{ is satisfied by } x^*\} \cup \{c\}.$$

We can easily verify that S is a harmless set and $|S \cap (V^-(x_i) \cup V^+(x_i) \cup E(x_i))| = 12$ and thus $|S \cap A| = 8m$ and then

$$opt(I) \geq |S| = 8m + opt(\phi) + 1 \quad (3)$$

Let S be a maximal harmless set for I . We first establish several useful observations.

Observation 1 *For each $i = 1, \dots, n$, the set S cannot contain vertices from both $V^-(x_i)$ and $V^+(x_i)$.*

The correctness follows from the fact that an edge-vertex cannot have both neighbors inside S .

Observation 2 *The set S cannot contain the vertex c' as well as any vertex \bar{c}'_j .*

The observation holds because any vertex \bar{c}'_j (resp. vertex c') is adjacent to the degree one vertex \bar{c}_j (resp. c).

Observation 3 *For each $i = 1, \dots, n$, the set S contains all vertices in $E(x_i)$.*

As to the correctness, consider a vertex $e \in E(x_i) \setminus S$ with its two neighbors v and v' . The set $S \cup \{e\}$ is also a harmless set since we know that v and v' are each adjacent to a vertex that is not contained in S according to Observation 2.

Observation 4 *The set S contains the vertex c .*

Indeed, using Observation 1 together with equations (1) and (2) we deduce that $N(c') \not\subseteq S \cup \{c\}$ and then $S \cup \{c\}$ is a harmless set.

Now, we show how to construct an assignment a_S for ϕ from the solution S such that $val(\phi, a_S) = |S| - 8m - 1$. If S contains for every $i = 1, \dots, n$ one of the sets $V^-(x_i)$ or $V^+(x_i)$ then $|S \cap A| = 8m$ and we can define the following assignment $a_S: x_i = 1 \Leftrightarrow |S \cap V^+(x_i)| \neq 0$. In this case, a clause-vertex \bar{c}_j is in S if and only if the corresponding clause is satisfied by a_S . Thus, the number of clauses satisfied by a_S is exactly $val(\phi, a_S) = |S| - 8m - 1$.

Assume now that $|S \cap A| < 8m$. We show that there exists another solution S' with $|S'| \geq |S|$ such that $|S' \cap A| = 8m$. We may assume that for each $i = 1, \dots, n$, we have either $|V^-(x_i) \cap S| \geq 1$ or $|V^+(x_i) \cap S| \geq 1$. Indeed, there always exists at least one vertex $v \in V^-(x_i) \cup V^+(x_i) \cap N(c')$ that can be added to S according to Observation 1 and equations (1) and (2). Moreover, we have $|V^-(x_i) \cap S| < 3$ and $|V^+(x_i) \cap S| < 3$ for some $i \in \{1, \dots, n\}$ since $|S \cap A| < 8m$ and, according to Observation 3, we know that S contains all vertices in $E(x_i)$ for every $i = 1, \dots, n$. Let $i \in \{1, \dots, n\}$ be such that $1 \leq |V^-(x_i) \cap S| < 3$ (the case $1 \leq |V^+(x_i) \cap S| < 3$ is symmetric). There must exist a vertex $v \in V^-(x_i) \setminus S$ which is either adjacent to c' or to a clause-vertex \bar{c}'_j . We first add v in S to get the new set S_v . If v is adjacent to a clause-vertex \bar{c}'_j such that $N(\bar{c}'_j) \subseteq S_v$ then remove \bar{c}_j from S_v . If v is adjacent to c' then we cannot have $N(c') \subseteq S_v$ since the way we added v into S is such that Observation 1 is preserved for the set S_v . We repeat the previous argument for each $v \in V^-(x_i) \setminus S_v$ and each $i = 1, \dots, n$ such that $1 \leq |V^-(x_i) \cap S_v| < 3$. Thus, we obtain a new solution S' such that $|S'| \geq |S|$ and $|S' \cap A| = 8m$. Similarly to the above case, we can obtain an assignment $a_{S'}$ such that $|S'| - val(\phi, a_{S'}) = 8m + 1$. In particular, if S' is an optimal solution, then

$$opt(\phi) \geq val(\phi, a_{S'}) = opt(I) - 8m - 1 \quad (4)$$

It follows from the inequalities (3) and (4) that $opt(I) - opt(\phi) = 8m + 1$ and then

$$opt(\phi) - val(\phi, a_{S'}) = opt(I) - |S'|$$

This completes the proof. \square

While the previous approximability results were essentially negative, we conclude this section with a polynomial-time approximation scheme for MAX HARMLESS SET on planar graphs. Notice that the problem is still NP-hard in this case since HARMLESS SET WITH UNANIMITY is equivalent to TOTAL DOMINATING SET which is NP-complete on planar graphs [27]. We leave as open question whether HARMLESS SET on planar graphs is fixed-parameter tractable with respect to k .

Theorem 16 MAX HARMLESS SET on planar graphs is in PTAS.

Proof: Given a planar embedding of an input graph, we consider the set of the vertices which are on the exterior face, they will be called *level 1 vertices*. By induction we define *level k* as the vertices which are on the exterior face when we have removed the vertices of levels smaller than k [3]. A planar embedding is *k -level* if it has no vertices of level greater than k . If a planar graph is k -level, it has a k -outerplanar embedding.

If we want to achieve an approximation within $1 + \varepsilon$, let us consider $k = 2(2 + \lceil \frac{1}{\varepsilon} \rceil)$. Let X_t be the set of vertices of level t and let H_i , $0 \leq i < k - 2$, be the graph obtained from G by considering the subgraphs formed by the set of vertices $\bigcup_{t+1 \leq j \leq t+k} X_j$, for $t \equiv i \pmod{(k-2)}$. The subgraph containing exactly $\bigcup_{t+1 \leq j \leq t+k} X_j$ is k -outerplanar, and so is H_i , too.

Since H_i is k -outerplanar, it has treewidth at most $3k - 1$ [9]. We construct graph H'_i from H_i by attaching a forbidden edge to each vertex on the boundary (that means vertices in $X_{t+1}, X_{t+2}, X_{t+k-1}, X_{t+k}$ with $t \equiv i \pmod{(k-2)}$). Thus, in each subgraph of H'_i the vertices in $X_{t+1}, X_{t+2}, X_{t+k-1}, X_{t+k}$ cannot be part of any harmless set.

On applying Theorem 11, we can efficiently determine an optimal harmless set in each subgraph of H'_i . Denote by S_i the union of these harmless sets. Clearly S_i is a harmless set on H_i .

Among S_0, \dots, S_{k-1} we choose the best solution that we denote S and we are going to prove that S is an $(1 + \varepsilon)$ -approximation of the optimal value on G . We can easily show that there is at least one r , $0 \leq r < k - 2$ such that at most $\frac{2}{k-2}$ of vertices in an optimal solution S_{opt} of G are on levels $X_{t+1}, X_{t+2}, X_{t+k-1}, X_{t+k}$ with $t \equiv r \pmod{(k-2)}$. This means that the solution S_r obtained by deleting the vertices from levels $X_{t+1}, X_{t+2}, X_{t+k-1}, X_{t+k}$ from S_{opt} will have at least $|S_{opt}|(1 - \frac{2}{k-2}) = \frac{k-4}{k-2} opt$ vertices. According to our algorithm, $|S| \geq |S_r| \geq \frac{opt}{1+\varepsilon}$.

The overall running time of the algorithm is k times what we need for graphs of treewidth at most k , that is $O(kt_{\max}^{O(k)} n) = n^{O(1/\varepsilon)}$ where t_{\max} is the maximum threshold. \square

6 Conclusion

In this paper, we introduced the HARMLESS SET problem. We established positive and negative results concerning its parameterized tractability and approximability. However, several questions remain open. For instance, we do not know if the problem is fixed-parameter tractable on general graphs with respect to the parameter treewidth and on planar graphs with respect to the solution size. Another interesting open question is whether HARMLESS SET WITH UNANIMITY is fixed-parameter tractable for parameter k when we ask to determine the existence of a harmless set of size at least $\lceil \frac{n}{3} \rceil + k$. Finally, another challenging question is to improve the factor-3 approximation of MAX HARMLESS SET WITH UNANIMITY.

Acknowledgments. We would like to thank the two anonymous referees for insightful comments that helped us to improve the presentation of the paper.

References

- [1] A. Aazami and K. Stilp. Approximation algorithms and hardness for domination with propagation. *SIAM Journal on Discrete Mathematics*, 23(3):1382–1399, 2009.
- [2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Prota. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.
- [3] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
- [4] C. Bazgan and M. Chopin. The robust set problem: parameterized complexity and approximation. In *37th International Symposium on Mathematical Foundations of Computer Science (MFCS '12)*, LNCS 7464, pages 136–147. 2012.
- [5] C. Bazgan, M. Chopin, A. Nichterlein, and F. Sikora. Parameterized approximability of influence in social networks. In *Proceedings of the 19th Annual International Computing and Combinatorics Conference (COCOON '13)*, LNCS 7936, pages 543–554. 2013.
- [6] O. Ben-Zwi, D. Hermelin, D. Lokshtanov, and I. Newman. Treewidth governs the complexity of Target Set Selection. *Discrete Optimization*, 8(1):87–96, 2011.
- [7] P. Berman and M. Karpinski. On some tighter inapproximability results (extended abstract). In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP '99)*, LNCS 1644, pages 200–209. 1999.
- [8] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
- [9] H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(12):1–45, 1998.
- [10] C. C. Centeno, M. C. Dourado, L. D. Penso, D. Rautenbach, and J. L. Szwarcfiter. Irreversible conversion of graphs. *Theoretical Computer Science*, 412(29):3693–3700, 2011.
- [11] M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67(4):654–685, 2003.
- [12] N. Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.
- [13] C.-Y. Chiang, L.-H. Huang, B.-J. Li, J. Wu, and H.-G. Yeh. Some results on the target set selection problem. *Journal of Combinatorial Optimization*, 25(4):702–715, 2013.
- [14] M. Chopin, A. Nichterlein, R. Niedermeier, and M. Weller. Constant thresholds can make target set selection tractable. *Theory of Computing Systems*, 55(1):61–83, 2014.

- [15] E. J. Cockayne, R. M. Dawes, and S. T. Hedetniemi. Total domination in graphs. *Networks*, 10(3):211–219, 1980.
- [16] F. Dehne, M. Fellows, H. Fernau, E. Prieto, and F. Rosamond. NONBLOCKER: Parameterized algorithmics for MINIMUM DOMINATING SET. In *Proceedings of SOFSEM 2006: Theory and Practice of Computer Science*, pages 237–245, 2006.
- [17] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141(12):109–131, 1995.
- [18] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag, 2013.
- [19] P. A. Dreyer, Jr. and F. S. Roberts. Irreversible k -threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615–1627, 2009.
- [20] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.
- [21] H. Fernau and J. A. Rodríguez-Velázquez. A survey on alliances and related parameters in graphs. *Electronic Journal of Graph Theory and Applications*, 2(1):70–86, 2014.
- [22] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [23] P. A. Golovach, J. Kratochvíl, and O. Suchý. Parameterized complexity of generalized domination problems. *Discrete Applied Mathematics*, 160(6):780–792, 2012.
- [24] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.
- [25] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002.
- [26] M. A. Henning and A. P. Kazemi. k -tuple total domination in graphs. *Discrete Applied Mathematics*, 158(9):1006–1011, 2010.
- [27] M. A. Henning and A. Yeo. *Total Domination in Graphs*. Springer-Verlag, 2013.
- [28] D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Co., 1997.
- [29] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 137–146, 2003.
- [30] S. Khanna, R. Motwani, M. Sudan, and U. V. Vazirani. On syntactic versus computational views of approximability. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS '94)*, pages 819–830, 1994.
- [31] T. Kloks. *Treewidth, Computations and Approximations*. Springer, 1994.

- [32] C. T. Nguyen, J. Shen, M. Hou, L. Sheng, W. Miller, and L. Zhang. Approximating the spanning star forest problem and its application to genomic sequence alignment. *SIAM Journal on Computing*, 38(3):946–962, 2008.
- [33] A. Nichterlein, R. Niedermeier, J. Uhlmann, and M. Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 3(2):233–256, 2013.
- [34] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- [35] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.
- [36] D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282:231–257, 2002.
- [37] T. V. T. Reddy and C. P. Rangan. Variants of spreading messages. *Journal of Graph Algorithms and Applications*, 15(5):683–699, 2011.
- [38] J. A. Telle. Complexity of domination-type problems in graphs. *Nordic Journal of Computing*, 1(1):157–171, 1994.
- [39] J. M. M. van Rooij, H. L. Bodlaender, and P. Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *17th Annual European Symposium on Algorithms (ESA '09)*, LNCS 5757, pages 566–577. 2009.
- [40] V. V. Vazirani. *Approximation Algorithms*. Springer, 2004.