The complexity of finding harmless individuals in social networks

Cristina Bazgan\textsuperscript{a,b}, Morgan Chopin\textsuperscript{a,*}

\textsuperscript{a} PSL, Universit\'e Paris-Dauphine, LAMSADE UMR 7243, Place du Mar\'echal de Lattre de Tassigny, 75775 Paris Cedex 16, France
\textsuperscript{b} Institut Universitaire de France, France

\textbf{ARTICLE INFO}

\textbf{Article history:}
Received 2 April 2013
Received in revised form 16 September 2014
Accepted 19 September 2014
Available online 18 October 2014

\textbf{Keywords:}
Approximation
Parameterized complexity
Harmless set
Treewidth
Social networks

\textbf{ABSTRACT}

In this paper, we introduce a domination-related problem called \textsc{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 \textsc{W[2]}-complete in general and \textsc{W[1]}-complete if all thresholds are bounded by a constant. Moreover, we prove that, if $P \neq NP$, the maximization version is not $n^{1+\varepsilon}$-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 \textsc{Harmless Set} is fixed-parameter tractable for parameter $k$ and the maximization version is \textsc{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 $(n-k)$-\textsc{Harmless Set} is fixed-parameter tractable for a large family of threshold functions.

\copyright 2014 Elsevier B.V. All rights reserved.

1. Introduction

The diffusion of information through social networks is a large and well-studied topic \cite{2}. One of the most well known problems that appear in this context is \textsc{Target Set Selection} introduced by Chen \cite{3} 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 \cite{4–6}. In this last example, the task for a company would be to advertise a 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 \cite{7,8,4,9,10}, polynomial-time approximability \cite{11,3}, parameterized approximability \cite{12}, and parameterized complexity \cite{13–15} 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

\cite{1} The preliminary results of this publication were presented at MFCS 2012 \cite{11}.

\* Corresponding author. Fax: +33 0144054091.
E-mail addresses: bazgan@lamsade.dauphine.fr (C. Bazgan), chopin@lamsade.dauphine.fr (M. Chopin).

http://dx.doi.org/10.1016/j.disopt.2014.09.004
1572-5286/© 2014 Elsevier B.V. All rights reserved.
Theorem 4 \[\text{FPT (Theorem 4)}\]

Not we establish parameterized intractability results for \((n - k)\)-Harmless Set along with our approximation results for \((n - k)\)-Harmless Set. Here, the parameter is \(k\) both for Harmless Set and \((n - k)\)-Harmless Set.

Theorem 6

We show that the parametric dual problem \(\text{Max Harmless Set}\) has a polynomial-time algorithm to solve it. In a very recent survey, Fernau and Rodríguez-Velázquez provided a connection between harmless set and alliances in graphs \cite{Fernau2019}.

Theorem 9

The parametric dual problem \(\text{Max Harmless Set}\) with various approximability properties is considered in the literature. The first results for this problem came from Henning et al. \cite{Henning2016}.

Unanimity

Theorem 8

We give a polynomial-time algorithm to solve \(\text{Max Harmless Set}\) with respect to the solution size in \cite{Bazgan2014}. The parameterized dual problem \((n - k)\)-Harmless Set is fixed-parameter tractable for a large family of threshold functions. In Section 4 we give a polynomial-time algorithm to solve \(\text{Max Harmless Set}\) for graphs of bounded treewidth. In Section 5 we establish that \(\text{Max Harmless Set}\) is not \(n^{1-\varepsilon}\)-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 \(\text{Max Harmless Set}\) is APX-complete. Moreover \(\text{Max Harmless Set}\) has a polynomial-time approximation scheme on planar graphs. Conclusions and open problems are given in Section 6.

Table 1

<table>
<thead>
<tr>
<th>Thresholds</th>
<th>Harmless Set</th>
<th>((n - k))-Harmless Set</th>
<th>Max Harmless Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>W[2]-complete (Theorem 4)</td>
<td>W[2]-hard</td>
<td>Not (n^{1-\varepsilon})-approx</td>
</tr>
<tr>
<td>Constant</td>
<td>W[1]-complete (Theorem 6)</td>
<td>FPT (Theorem 9)</td>
<td>Not (n^{1-\varepsilon})-approx (Theorem 13)</td>
</tr>
<tr>
<td>Majority</td>
<td>W[1]-hard (Theorem 6)</td>
<td>FPT (Theorem 9)</td>
<td>Not (n^{1-\varepsilon})-approx (Theorem 13)</td>
</tr>
<tr>
<td>Unanimity</td>
<td>FPT (Theorem 8)</td>
<td>W[2]-hard(^a)</td>
<td>APX-complete (Theorem 15)</td>
</tr>
</tbody>
</table>

\(^a\) The result marked 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 \cite{Bazgan2014}.

Fig. 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.

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 a harmless set as a converse notion of a 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 \(t : V \to \mathbb{N}\), and an integer \(k\), determine whether there exists a harmless set of size at least \(k\) (see Fig. 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 raises 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 \cite{Dehne2016} 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 have 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. \cite{Dehne2016}. 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 \cite{Dehne2016} or a spanning star forest \cite{Telle2015} 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 \(\ell\)-tupel Total Dominating Set and introduced by Henning et al. \cite{Henning2016}. The difference with the original version is that a vertex \(v\) is now dominated if and only if at least \(\ell\) of its neighbors are in \(S\). One can observe that an \(\ell\)-tupel 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 \cite{Fernau2019}. Finally, we can relate our problem to \((\sigma, \rho)\)-Dominating Set introduced by Telle \cite{Telle2015}. Given a graph \(G = (V, E)\), two sets \(\sigma, \rho\) of non-negative integers, and an integer \(k\), the goal is to find a \((\sigma, \rho)\)-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 a \((\sigma, \rho)\)-dominating set of size \(k\) \cite{Bazgan2014} where \(\sigma = \rho = \{0, \ldots, c - 1\}\).

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 parameterual dual problem \((n - k)\)-Harmless Set 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 \(n^{1-\varepsilon}\)-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.
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 $t : V \to \mathbb{N}$ 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 Fig. 1). In each figure, we indicate the thresholds inside the vertices. We define in the following the problems we study in this paper.

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.

If the threshold function is defined by $t(v) = d(v)$ for every $v \in V$, then we add the suffix WITH UNANITY 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 [23–25]. 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 to $f$. 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 decidable). 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 $\text{FPT} \subseteq W[1] \subseteq W[2] \cdots \subseteq \text{XP}$. Informally speaking, a problem 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 $\text{opt}_A(I)$ the optimum value of $I$ and by $\text{val}(I, S)$ the value of a feasible solution $S$ of $I$. For a function $\rho > 1$, an algorithm is a $\rho$-approximation for a maximization problem $A$ if for any instance $I$ of the problem it returns a solution $S$ such that $\text{val}(I, S) \geq \frac{\text{opt}_A(I)}{\rho|I|}$. We say that a maximization problem is constant approximable if, for some constant $\rho > 1$, there exists a polynomial-time $\rho$-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 [26]). 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 $\text{opt}_B(I') \leq \alpha \cdot \text{opt}_A(I)$,
2. $g$ maps each solution $S'$ of $I'$ into a solution $S$ of $I$ such that $|\text{val}(I, S) - \text{opt}_A(I)| \leq \beta \cdot |\text{val}(I', S') - \text{opt}_B(I')|$.

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 $B \in \text{PTAS}$, then $A \in \text{PTAS}$. 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 [27]). 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 $\text{opt}_B(I')$ and $\text{opt}_A(I)$ are related by a polynomial factor, i.e. there exists a polynomial $p$ such that $\text{opt}_B(I') \leq p(|I|) \cdot \text{opt}_A(I)$,
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 $\varepsilon(I, S) = \max \left\{ \frac{|\text{val}(I, S) - \text{opt}_A(I)|}{\text{opt}_A(I)}, \frac{|\text{val}(I', S') - \text{opt}_B(I')|}{\text{opt}_B(I')} \right\} - 1$.

For more details about approximation the reader is referred to [28–30].

Tree decomposition and treewidth. A tree decomposition $\mathcal{T} = (T, \mathcal{X})$ of a graph $G = (V, E)$ consists of a tree $T = (X, F)$ with a node set $X$ and an edge set $F$, and a set system $\mathcal{X}$ over $V$ whose members $H_x \in \mathcal{X}$ 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 $v \in H_x \cap H_x'$ holds for every node $x$ of the (unique) $x' \rightarrow x''$ 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 $\mathcal{T} = (T, \mathcal{X})$ 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 $\mathcal{T} = (T, \mathcal{X})$ of a graph can be transformed in linear time into a so-called nice tree decomposition $\mathcal{T}' = (T', \mathcal{X}')$ with $w(\mathcal{T}') = w(\mathcal{T})$, $\mathcal{X}' = O(\mathcal{X})$ and with $H_x \neq \emptyset$ for all $H_x \in \mathcal{X}$ 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 $H_y' = H_z' = H_x' \cup H_x'$ (x is called $\text{join node}$) with $H_y', H_z', H_x' \in \mathcal{X}'$.
3. If a node $x$ has just one child $y$, then $H_y' \subset H_x'$ (x is called $\text{forget node}$) or $H_x' \subset H_y'$ (x is called $\text{insert node}$) and $||H_x' - |H_y'|| = 1$ with $H_y', H_x' \in \mathcal{X}'$.

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 $uw$ 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.

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 [32] 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 \(V = \{v_1, \ldots, v_n\}\). 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 Fig. 2). We create \(n + 1\) tapes denoted by \(T_0, T_1, \ldots, T_n\). The tape alphabet is \(V \cup \{\square\}\) plus the blanks 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 the 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 \(i = 1, \ldots, k + 1\), the Turing machine writes a symbol \(\times\) on each tape \(T_i\) and moves \(T_i\)'s head one step to the right if \(v_i\) has a threshold greater than 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 the 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 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 [23].

**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 \(\max\{k - d_G(v), 1\}\) 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 Fig. 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 \(\lor\)-gate connected to the \(\neg\)-gates of inputs in \(S'\). Finally, add a large \(\land\)-gate connected to every \(\lor\)-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 \(C\). \(\square\)
Theorem 4

We now prove the W[1]-hardness. Let \( G \). The vertices in the gray box form a clique.

Fig. 4. Illustration of the reductions (1) and (2) from Clique as described in Theorem 6. The vertices in the gray box form a clique.

We establish the W[1]-hardness of Harmless Set by an fpt-reduction from the Clique problem [33] 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 \( u e_{uv} \) 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 Fig. 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 \( u e_{uv} \) 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 Fig. 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 \( v \in V' \backslash V \), then there is always a vertex \( w \in V' \backslash V - \{u, v\} \) 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 \( G \). □

It is interesting to note that the ratio between the number \( n_0 \) 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 \( \frac{n_0}{n} \geq 0 \) 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.

**Theorem 7** (Cockayne et al. [34]). 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 size 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 $G'$ 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 at 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 $3k'$. \hfill \Box

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 the first one as an open question.

**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 [22]. 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 $t(v) = [\alpha_v d(v)^{\beta_v} + \gamma_v]$ 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

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

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

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

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: $x' = \epsilon x$ if and only if $x \geq (1/\epsilon)\frac{1}{1-\epsilon}$ for any $x \geq 1$ and $\epsilon \in (0, 1)$. 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 $(G', t', k)$. \hfill \Box
4. Algorithms for trees and tree-like graphs

In this section we establish an $t_{\text{max}}^{O(\omega)} \cdot n$-time algorithm for MAX HARMLESS SET and a $k^{O(\omega)} \cdot n$-time algorithm for HARMLESS SET where $t_{\text{max}}$ is the maximum threshold and $\omega$ the width of a given tree decomposition of the input graph. We first describe an $O(\log(t_{\text{max}}) \cdot n)$-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 $O(\log(t_{\text{max}}) \cdot n)$ 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 $\bot$.

As a preliminary step, we set $F(p(v)) = \bot$ whenever $t(v) = 1$ for every vertex $v \neq r$. Thus a vertex $v$ is such that $F(v) = \bot$ 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 $I_v[1] = E_v[1] = \bot$. 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 $c_1, c_2 \in C_1(v, b_1, b_2)$ 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:

$$I_v[b] = \begin{cases} \bot & \text{if } F(v) = \bot \\ \{v\} \cup \bigcup_{c \in C_1(v, b, 1)} I_c[1] \cup \bigcup_{c \in C_2(v, b, 1)} E_c[1] & \text{if } t(v) = 1 \text{ and } b = 1 \\ \bot \cup \bigcup_{c \in C_1(v, b, 0)} I_c[0] \cup \bigcup_{c \in C_2(v, b, 0)} E_c[0] & \text{otherwise.} \end{cases}$$

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

In the above equations, we adopt the convention that $Q \cup \bot = \bot$ 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 $I_v[0], E_v[0], I_v[1], \text{ and } E_v[1]$.

We now prove Eq. (1) (Eq. (2) is proved using similar arguments). In the first case ($F(v) = \bot$) there exists a child $c$ of $v$ such that $t(c) = 1$ and thus $v$ cannot be part of any harmless set thus $I_v[b] = \bot$. In the second case, we set $I_v[b] = \bot$ 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 $C(v, 1, 1) = \{c \in C(v) : x^*_c = 1\}$. 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 leaves’ tables can be done in $O(n)$ time. Furthermore, the number of steps performed in each non-leaf vertex is $O(\log(t_{\text{max}}) \cdot |C(v)|)$. Indeed, partitioning the set $C(v)$ (Step 2) requires $O(\log(t_{\text{max}}) \cdot |C(v)|)$ time and both Step 1 and Step 3 use $O(|C(v)|)$ time. Overall, the running time is $O(\sum_{v \in V} \log(t_{\text{max}}) \cdot |C(v)|) = O(\log(t_{\text{max}}) \cdot n)$. This completes the proof. □
Now, we present the algorithm for solving MAX HARMLESS SET on general graphs.

**Theorem 11.** Given a tree decomposition of width \( \omega \) of a graph \( G \), a maximum harmless set can be computed in time \( t_{\text{max}}^\omega \cdot n \) where \( t_{\text{max}} \) 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 \( \omega \). 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 \( \bigcup_{y \in T_x} \mathcal{H}_y \). 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 \( A_x[t, c] \) where \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_x}, \ c \in \{0, 1\}^{\mathcal{H}_x}, \) and \( A_x[t, c] \) corresponds to a maximum harmless set in \( G_x \) whose intersection with \( H_x \) is exactly \( \{v \in H_x : c(v) = 1\} \) and by imposing the threshold \( t(v) = t(v) \) for every \( v \in H_x \). We set \( A_x[t, c] = \perp \) if no such harmless set is possible. During the computation, we may interrogate the table \( A_x \) with \( t(v) \leq 0 \) 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 \( H_y = H_z = H_x \). 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 \( B = N_c(H_x) \backslash V_x \). Notice that, when computing \( A_x \), we do not know which 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 \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_x} \). 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 \( t \) to \( H_y \) (resp. \( H_z \)) for some \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_x} \). 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 \( t \). Indeed, consider a vertex \( u \in H_x \). It may happen that \( u \) has less than \( t(u) \) neighbors in \( S_y \) and \( S_z \) but more than \( t(u) \) in \( S_y \cup S_z \). To overcome this situation, we have to consider the union of a pair of optimal solutions \( S_y, S_z \) for each possible threshold value \( t_i \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_y} \) and \( t_i \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_z} \) of the vertices in \( H_y \) and \( H_z \), respectively, with \( t_1 + t_2 = t(t_1 + t_2) \) is the function defined as \( (t_1 + t_2)(u) = t(t_1(u) + t_2(u)) \) for every \( u \in H_x \).

The other problem is whenever we update 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 \( G_x \) might have a number of neighbors in \( H_x \cap (S_y \cup S_z) \) that sums over its threshold. We solve this issue using the function \( c \). According to the definition of \( A_x[t, c] \), 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 \( \{c(v) = 1\} \) may be adjacent to some vertex \( u \) in \( H_x \) and thus affects \( t(u) \). Since we consider \( H_y \) and \( H_z \) separately, we count the vertex \( v \) once for \( t_i(u) \) and a second time for \( t_i(u) \). This problem can be overcome by simply increasing the thresholds of \( u \) in both \( H_y \) and \( H_z \) so as to balance this overcounting.

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

**Algorithm.** We denote by \( f \oplus ((a, b)) \) the extended function defined as \( (f \oplus ((a, b)))(x) = f(x) \) if \( x \neq a \) and \( (f \oplus ((a, b)))(a) = b \). 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 \( T, t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_x} \) and \( c \in \{0, 1\}^{\mathcal{H}_x} \). Let \( S = \{v \in H_x : c(v) = 1\} \).

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

**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 \( H_y = H_z = \{u\} \). Following the above discussion, we update the table \( A_x \) as follows. For all \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_y} \) and \( c \in \{0, 1\}^{\mathcal{H}_y} \) and \( i = 1, \ldots, t_{\text{max}} \)

\[
\begin{align*}
A_x[t \oplus (u, i), c \oplus (u, 0)] &= \{A_y[t, c] \text{ if } A_y[t, c] \text{ is a harmless set in } G_x \text{ with } t(u) = i \\
&\cup \perp \text{ otherwise} \\
A_x[t \oplus (u, i), c \oplus (u, 1)] &= \{A_y[t, c] \cup \{u\} \text{ if } A_y[t, c] \cup \{u\} \text{ is a harmless set in } G_x \text{ with } t(u) = i \\
&\cup \perp \text{ otherwise} 
\end{align*}
\]

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

**Case 2 (forget node).** Suppose that \( x \) is a forget node with child \( y \) such that \( H_y = H_z \). Let \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_y} \). Notice that vertex \( u \) has its neighbors entirely contained in \( G_y \). Hence, the maximum harmless set for \( G_x \) where \( t(v) = t(u) \) for all \( v \in H_x \) is exactly the maximum harmless set for \( G_y \) where \( t(v) = t(u) \) 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 \( t \in \{1, \ldots, t_{\text{max}}\}^{\mathcal{H}_y} \) and \( c \in \{0, 1\}^{\mathcal{H}_y} \)

\[
A_x[t, c] = \max_{i \in \{0, 1\}} \{A_y[t \oplus (u, t(u)), c \oplus (u, i)]\}.
\]
Case 3 (join node). Suppose that \( x \) is a join node with children \( y \) and \( z \) such that \( H_k = H_r = H_z \). According to the above discussion, we update the table \( A_k \) as follows. For all \( t \in \{1, \ldots, t_{\text{max}}\}^{H_k} \) and \( c \in \{0, 1\}^{H_k} \), perform the following steps:

- Set \( A(u) = \{(v \in H_k : u \in N(v) \text{ and } c(v) = 1)\} \) for all \( u \in H_k \).
- \((t_1, t_2) = \arg\max_{t_1, t_2 \in \{1, \ldots, n\}^{H_k}} |A_2[t_1 + e, c] \cup A_2[t_2 + e, c]| \).
- \( A[t, c] = A[t_1 + e, c] \cup A[t_2 + e, c] \).

Final step. The optimal solution is then \( \max_{c \in \{0, 1\}^{H_k}} |A[t, c]| \) where \( r \) is the root of \( T \) and \( t_r(v) = t(v) \) for all \( v \in H_r \). As to the running time observe that enumerating all possible functions \( t \) and \( c \) takes \( O(n) \) 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 \( t_{\text{max}} = k + 1 \). We can apply the algorithm of Theorem 11 on \( (G', t') \) and then decide whether the given solution is greater than or equal to \( k \). This gives us the following result.

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

Since a tree decomposition of width \( tw(G) \) of a graph \( G \) can be found in \( fpt \)-time with respect to \( tw(G) \) [35], 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 [36].

### 5. Approximability

We first observe that \( \text{Max Harmless Set} \) 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 \( n^{1/2 - \varepsilon} \) 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 \( \text{opt}(I') = \text{opt}(G) \) 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 \( n^{1/2 - \varepsilon} \) for any \( \varepsilon > 0 \) unless \( NP = ZPP \) [37] 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 \( \Delta \) 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 \( N_C(v) \not\subseteq S \). 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, \( \text{opt}(G) \leq |V_1| + |V_{\geq 3}| \). It follows that \( |S| \geq \text{opt}(G)/3 \). \( \square \)

**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 E 2SAT-3 proved APX-hard in [38] and is defined as follows: given a CNF formula \( \phi \) 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 \( m = 3n/2 \).
We can easily verify that $S_{16}^{\text{opt}}(\phi)$

Observation 1. The correctness follows from the fact that an edge-vertex cannot have both neighbors inside $V^-(x_i)$.

Observation 2. For every clause $c_j$ in $\phi$ add two adjacent clause-vertices $c_j'$ and $\overline{c}_j'$.

Observation 3. 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 [38], we may assume that each variable of $\phi$ 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.

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

and

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

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

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

$$\text{opt}(I) \leq 18\text{opt}(\phi)$$

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

Moreover, let $x^*$ be an optimal assignment for $\phi$ and let

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

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

$$\text{opt}(I) \geq |S| = 8m + \text{opt}(\phi) + 1.$$ (3)

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

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

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

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

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

As to the correctness, consider a vertex $e \in E(x_i) \backslash S$ 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 Eqs. (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 $a_c$ for $\phi$ from the solution $S$ such that $\text{val}(\phi, a_c) = |S| = 8m - 1$. If $S$ contains for each $i = 1, \ldots, n$ one of the sets $V^-(x_i) \cup V^+(x_i)$, then $|S \cap A| = 8m$ and we can define the following assignment $a_c: x_i = 1 \iff |S \cap V^+(x_i)| \neq 0$. In this case, a clause-vertex $c_i$ is in $S$ if and only if the corresponding clause is satisfied by $a_c$. Thus, the number of clauses satisfied by $a_c$ is exactly $\text{val}(\phi, a_c) = |S| - 8m - 1$.

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 $i = 1, \ldots, n$, we have either $V^-(x_i) \cap S \not\subseteq S$ according to Observation 1 and Eqs. (1) and (2). Moreover, we have $|V^-(x_i) \cap S| < 3$ and $|V^+(x_i) \cap S| < 3$ for some $i \in \{1, \ldots, n\}$ since $|S \cap A| < 8n$ and, according to Observation 3, we know that $S$ contains all vertices in $E(x_i)$ for every $i = 1, \ldots, n$. Let $i \in \{1, \ldots, n\}$ 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 $x_i \in V^-(x_i) \setminus S$ which is either adjacent to $c_i$ or to a clause-vertex $c_i$. We first add $x_i$ in $S$ to get the new set $S_1$. If $x_i$ is adjacent to a clause-vertex $c_i$ such that $N(c_i) \subseteq S_1$, then remove $c_i$ from $S_1$. If $x_i$ is adjacent to $c_i$, then we cannot have $N(c_i) \subseteq S_1$, since the way we added $x_i$ into $S$ is such that Observation 1 is preserved for the set $S_1$. We repeat the previous argument for each $i = 1, \ldots, n$ such that $1 \leq |V^-(x_i) \cap S_1| < 3$. 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 $a_c$ such that $|S'| = \text{val}(\phi, a_c) = 8m + 1$. In particular, if $S'$ is an optimal solution, then

$$\text{opt}(\phi) \geq \text{val}(\phi, a_c) = \text{opt}(l) - 8m - 1.$$  

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

$$\text{opt}(\phi) - \text{val}(\phi, a_c) = \text{opt}(l) - |S'|.$$  

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 [16]. We leave as an 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 $k$ [39]. 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 $k = 2(2 + \lceil 1/\varepsilon \rceil)$. Let $X_i$ be the set of vertices of level $i$ and let $H_1, 0 \leq i < k - 2$, be the graph obtained from $G$ by considering the subgraphs formed by the set of vertices $\bigcup_{r+1 \leq j \leq i} X_j$ for $i \equiv (\text{mod } (k - 2))$. The subgraph containing exactly $\bigcup_{r+1 \leq j \leq i} X_j$ is $k$-outerplanar, and so is $H_1$.

Since $H_1$ is $k$-outerplanar, it has treewidth at most $3k - 1$ [40]. We construct graph $H_i'$ from $H_i$ by attaching a forbidden edge to each vertex on the boundary (that means vertices in $X_{i+1}, X_{i+2}, X_{i+k-1}, X_{i+k}$ with $i \equiv 1 \text{ (mod } (k - 2))$). Thus, in each subgraph of $H_i'$ the vertices in $X_{i+1}, X_{i+2}, X_{i+k-1}, X_{i+k}$ cannot be part of any harmless set.

By 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 $S_0, \ldots, S_{k-1}$ we choose the best solution that we denote as $S$ and we are going to prove that $S$ is an $(1 + \varepsilon)$-approximation of the optimal value $O$. 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 are on levels $X_{r+1}, X_{r+2}, X_{r+k-1}, X_{r+k}$ with $i \equiv r \text{ (mod } (k - 2))$. This means that the solution $S_{r}$ obtained by deleting the vertices from levels $X_{r+1}, X_{r+2}, X_{r+k-1}, X_{r+k}$ from $S_i$, will have at least $|S_{opt}|(1 - \frac{2}{k-2}) = \frac{k-2}{k-2} |S_i|$ vertices. According to our algorithm, $|S| \geq |S_{opt}| \geq \frac{1}{1 + \varepsilon} |S_i|$. Consequently, the overall running time of the algorithm is $k$ times what we need for graphs of treewidth at most $k$, that is $O(kt_{\text{max}}^{O(k)} n) = n^{O(1/\varepsilon)}$ where $t_{\text{max}}$ 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 are asked to determine the existence of a harmless set of size at least $\lceil \frac{k}{2} \rceil + k$. 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