Constant Thresholds Can Make Target Set Selection Tractable*

Morgan Chopin[1**], André Nichterlein[2], Rolf Niedermeier[2], and Mathias Weller[2***]

[1] LAMSADE, Université Paris-Dauphine, France
chopin@lamsade.dauphine.fr
[2] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
{andre.nichterlein, rolf.niedermeier, mathias.weller}@tu-berlin.de

Abstract. Target Set Selection, which is a prominent NP-hard problem occurring in social network analysis and distributed computing, is notoriously hard both in terms of achieving useful approximation as well as fixed-parameter algorithms. The task is to select a minimum number of vertices into a "target set" such that all other vertices will become active in course of a dynamic process (which may go through several activation rounds). A vertex, which is equipped with a threshold value t, becomes active once at least t of its neighbors are active; initially, only the target set vertices are active. We contribute further insights into islands of tractability for Target Set Selection by spotting new parameterizations characterizing some sparse graphs as well as some "cliquish" graphs and developing corresponding fixed-parameter tractability and (parameterized) hardness results. In particular, we demonstrate that upper-bounding the thresholds by a constant may significantly alleviate the search for efficiently solvable, but still meaningful special cases of Target Set Selection.

1 Introduction

The NP-hard graph problem Target Set Selection (TSS) is defined as follows: Given an undirected graph $G = (V, E)$ where each vertex $v \in V$ is assigned a positive integer threshold value $\mathrm{thr}(v)$, the task is to find a minimum-cardinality target set $S \subseteq V$. A vertex set $S \subseteq V$ is called a *target set* of G if it "activates" *all* vertices in G in a dynamic process where a vertex v gets activated once at least $\mathrm{thr}(v)$ many of its neighboring vertices are activated. Initially, only the vertices in S are active. TSS generalizes well-known graph problems such as Dominating Set with thresholds [14], Vector Dominating Set [25], k-Tuple Dominating Set [18] (all these variants allow for only one "activation round"), Vertex Cover [5] (where the threshold value equals the vertex degree), Irreversible k-Conversion Set [10] and r-Neighbor

* Major part of this work was done while all authors were at TU Berlin.
** Main work done during a three-month visit of TU Berlin supported by DAAD.
*** Supported by the DFG, project DARE (NI 369/11).

To appear in the Proceedings of the The 1st Mediterranean Conference on Algorithms (MedAlg)*, Ein-Gedi, Israel, December 2012, Lecture Notes in Computer Science.*
© *Springer*

Bootstrap Percolation [1] (where the threshold of each vertex is k or r, respectively), and so-called dynamic monopolies [24] (where the threshold of a vertex v with degree $\deg(v)$ equals $\lceil \deg(v)/2 \rceil$—in the following this condition is referred to as *majority* thresholds). Besides being a problem of considerable graph-theoretic interest, TSS is also motivated by applications in areas such as social network analysis [5, 17] and distributed computing [24]. Indeed, different research communities using different names for describing the same or similar concepts, some work has been done independently from each other.

Since previous work has shown that TSS is computationally very hard [2, 5, 22], it is a natural approach to search for practically relevant, but computationally tractable special cases. We contribute to this line of research by starting from the following: While TSS is linear-time solvable both on trees [5] and on cliques [22, 26], it turns hard if the treewidth is unbounded [2] (more specifically, it is W[1]-hard with respect to the parameter treewidth of the graph) and it is NP-hard on graphs with diameter two [22] (cliques are exactly the diameter-one graphs). In this work, we focus on parameterizations measuring the distance from being a tree or forest and parameterizations measuring the distance from being a clique or cluster graph. Along these lines, we are particularly interested in the role of the allowed thresholds and one of our main conclusions is that bounding the thresholds by a constant (independent of the vertex degree) may be decisive in order to gain (fixed-parameter) tractability. This is of interest since in several applications (such as influence spreading in social networks) it is conceivable that constant thresholds suffice to model the underlying application scenarios.[3]

Previous Work. We focus on previous results that have direct relation to our work and refrain from discussing the history of work on TSS (a more thorough review of previous work can be found in [22]).

Chen [5] showed hardness of approximation for TSS, and Ben-Zwi et al. [2] obtained parameterized hardness results, particularly with respect to the parameter "treewidth".[4] Recently, further parameterized complexity studies for the structural graph parameters "diameter", "cluster editing number", "vertex cover number", and "feedback edge set number" have been undertaken [22]. Moreover, polynomial-time algorithms for TSS restricted to special graph classes including chordal graphs and block-cactus graphs have been developed [4, 6, 26].

The role of the threshold values resp. threshold functions has been studied in some publications. For instance, Dreyer and Roberts [10] showed NP-hardness for TSS when all vertices have a threshold of t, $t \geq 3$. Centeno et al. [4] and Chiang et al. [6] exploited threshold values being upper-bounded by two to develop polynomial-time algorithms for TSS on chordal graphs. Most interesting in our context, however, is the result of Ben-Zwi et al. [2] who showed that TSS is

[3] For instance, independent of my total number of friends it may suffice that at least five of my friends (that is, neighbor vertices) in a social network buy a certain product to convince me about the product's usefulness.

[4] Indeed, they also showed that TSS is polynomial-time solvable on constant-treewidth graphs. However, the degree of the polynomial depends on the treewidth.

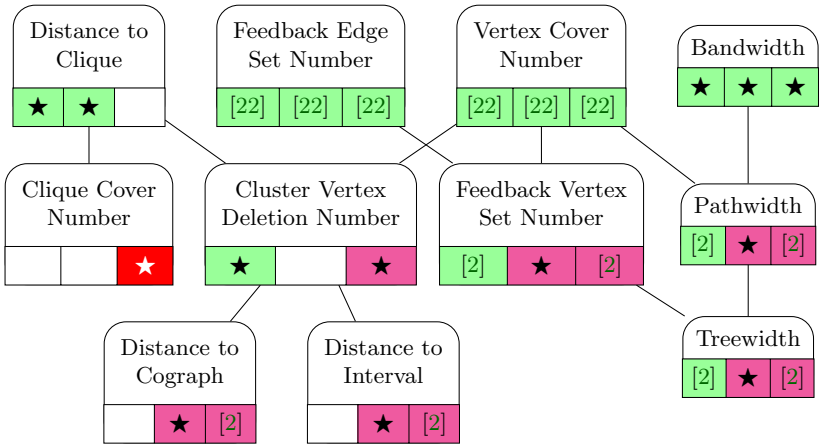

Fig. 1. Overview of the relations between structural graph parameters and our results for TSS. An edge from a parameter α to a parameter β below of α means that β can be upper-bounded in a linear function in α. The three rectangles below each parameter indicate the known results for TSS with (from left to right:) constant, majority, and general threshold function. The white star at the "clique cover number" means NP-hard for constant values of this parameter, dark violet background means W[1]-hard, light green background means FPT, and white background indicates an open question. Results marked with a star are obtained in this paper.

W[1]-hard with respect to the treewidth in case of unbounded threshold values whereas they showed it to be fixed-parameter tractable for the same parameter once the threshold values are bounded by a constant.

Our Contributions. Starting from the efficient solvability of TSS on trees and cliques [5, 22, 26], we also investigate to what extent efficient algorithms can be obtained for more general graph classes. On the one hand, we generalize from trees, thus in a way following previous work [2, 22] and consider further parameters measuring tree-likeness or sparseness, and, on the other hand we spot several parameters measuring distance to "cliquish" graphs. In both lines, we put particular emphasis on how the allowed threshold functions (arbitrary versus majority versus constant) influence computational complexity.[5] Our findings, which are pictorially presented (including the relations between parameters) in Figure 1, read as follows.

We start with the "sparse setting". For majority thresholds, we show that W[1]-hardness results for parameters such as "feedback vertex set number" and "pathwidth" for general threshold functions (which are due to Ben-Zwi et al. [2])

[5] Notably, all our positive results for constant thresholds generalize to the case that the maximum threshold $t_{\max}$ is given as an additional parameter. However, to keep matters simple and in accordance with previous work we focus on constant thresholds.

extend to the case of majority thresholds. Conversely, the very same parameterizations lead to fixed-parameter tractability results in case of constant threshold values [2]. Further, we briefly indicate that TSS is fixed-parameter tractable for the parameter bandwidth[6] even in case of arbitrary threshold functions.

Our main results are related to the "cliquish setting". The most central result here is that, with respect to the parameter "cluster vertex deletion number" (that is, the minimum number of vertices to delete from a graph to transform it into a union of disjoint cliques [15]), TSS is W[1]-hard for general thresholds but becomes fixed-parameter tractable in case of constant thresholds (the case of majority thresholds is open). For the larger (thus "weaker" [19]) parameter "distance to clique" (the minimum number of vertices to delete to make a graph a clique), however, TSS is fixed-parameter tractable for both constant and majority thresholds whereas this is open for general thresholds. Finally, for the parameter "clique cover number" (the minimum number of cliques needed to cover all vertices of a graph) we show NP-hardness even for parameter value two (rendering fixed-parameter tractability very unlikely) whereas the (parameterized) complexity is open in case of majority and constant thresholds.

Several proof details (particular concerning hardness proofs) are deferred to a full version of the paper.

2 Preliminaries and Parameter Identification

Preliminaries. We use standard graph-theoretic notation. For graphs $G = (V, E)$, we use $n := |V|$ and $m := |E|$. We omit the index of the neighborhood $N_G(v)$ or degree $\deg_G(v)$ of a vertex v if G is clear from the context. To formally define Target Set Selection, consider a graph $G = (V, E)$ and a function $\mathrm{thr} : V \to \mathbb{N} \cup \{0\}$. For a vertex set $S \subseteq V$, we define the set of vertices that are *activated by S in the ith round* as $\mathcal{A}^i_{G,\mathrm{thr}}(S)$ with

$$\mathcal{A}^0_{G,\mathrm{thr}}(S) := S \text{ and}$$
$$\mathcal{A}^{j+1}_{G,\mathrm{thr}}(S) := \mathcal{A}^j_{G,\mathrm{thr}}(S) \cup \{v \in V \mid |N(v) \cap \mathcal{A}^j_{G,\mathrm{thr}}(S)| \geq \mathrm{thr}(v)\}.$$

We call $r(S) := \max\{i \mid \mathcal{A}^{i-1}_{G,\mathrm{thr}} \neq \mathcal{A}^i_{G,\mathrm{thr}}\}$ the *number of activation rounds* and say that S is a *target set* for (G, thr) if $\mathcal{A}^{r(S)}_{G,\mathrm{thr}}(S) = V$. We can now formally define the central problem of this work:

Target Set Selection (TSS)
Input: An undirected graph $G = (V, E)$, a threshold function $\mathrm{thr} : V \to \mathbb{N} \cup \{0\}$ and an integer $k \geq 0$.
Question: Is there a target set $S \subseteq V$ for G with $|S| \leq k$?

We denote the maximum threshold of an instance (G, thr) by $t_{\max}(G, \mathrm{thr}) := \max\{\mathrm{thr}(v) \mid v \in V(G)\}$. Again, we omit (G, thr) if it is clear from the context.

[6] A graph with *bandwidth* k has a linear arrangement of its vertices $v_1, \ldots, v_n$ such that the length $|i - j|$ of each edge $\{v_i, v_j\}$ is at most k.

Parameterized Complexity. One dimension of a parameterized problem is the input size s, and the other one is the *parameter* (usually a positive integer). A parameterized problem is called *fixed-parameter tractable* (fpt) with respect to a parameter k if it can be solved in $f(k) \cdot s^{O(1)}$ time, where f is a computable function only depending on k. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction* [3, 13]. Here, the goal is to transform a given problem instance I with parameter k in polynomial time into an equivalent instance I' with parameter $k' \leq k$ such that the size of I' is upper-bounded by some function g only depending on k. If this is the case, we call I' a (problem) *kernel* of size $g(k)$. Usually, this is achieved by applying polynomial-time executable data reduction rules. We call a data reduction rule $\mathcal{R}$ *correct* if the new instance I' that results from applying $\mathcal{R}$ to I is a yes-instance if and only if I is a yes-instance. An instance is called *reduced* with respect to some data reduction rule if further application of this rule has no effect on the instance. The whole process is called *kernelization.*[7]

Downey and Fellows [9] developed a parameterized theory of computational complexity to show (presumable) fixed-parameter intractability by means of *parameterized reductions.* A parameterized reduction from a parameterized problem P to another parameterized problem P' is a function that, given an instance (x, k), computes in $f(k) \cdot s^{O(1)}$ time an instance (x', k') (with k' only depending on k) such that (x, k) is a yes-instance of P if and only if (x', k') is a yes-instance of P'. The basic complexity class for fixed-parameter intractability is called $W[1]$ and there is good complexity-theoretic reason to believe that $W[1]$-hard problems are not fpt [9, 12, 23].

Parameter Identification. Fixed-parameter algorithms work best if the parameter they are designed for is small in practice. TSS having many applications on social networks [11], it is natural to extract small parameters from typical properties of social networks.

When the network models friendships for example, we expect the network to be made up of multiple cliques (or otherwise dense substructures) that overlap. This motivates considering the number of cliques needed to cover all vertices [16] (the "clique cover number") or the number of vertices to remove to obtain a clique (the "distance to clique"). As the latter parameter is somewhat restrictive, we also considered the number of vertices to delete in order to obtain a collection of disjoint cliques (the "cluster vertex deletion number"). Recently, the cluster vertex deletion number was also used to parameterize problems related to coloring and hamiltonicity [8].

In some applications, we deal with very sparse social networks, for instance networks modeling sexual contacts [11, Chap. 2, Fig. 2.7]. In these cases, parameters related to the sparseness of the input graph are interesting. Among them, we consider the number of vertices to remove to obtain an edgeless graph ("vertex cover number"), the number of edges or vertices to remove to obtain a forest

[7] It is well-known that a parameterized problem is fixed-parameter tractable if and only if it has a problem kernel.

("feedback edge set number" and "feedback vertex set number") as well as some graph width parameters (treewidth, pathwidth, bandwidth). For definitions of treewidth, pathwidth, cographs, and interval graphs see for example Diestel's book [7].

Data Reduction. We use the following two data reduction rules throughout this work.

If the threshold of a vertex exceeds its degree, it cannot be activated by its neighbors and, hence, the vertex is part of any target set. Moreover, we consider threshold-0 vertices as already active.

Reduction Rule 1 ([22, Reduction Rule 1]) *Let $G = (V, E)$ and $v \in V$. If $\mathrm{thr}(v) > \deg(v)$, then delete v, decrease the threshold of all its neighbors by one and decrease k by one. If $\mathrm{thr}(v) = 0$, then delete v and decrease the thresholds of all its neighbors by one.*

If the instance is reduced with respect to Reduction Rule 1, then every degree-one vertex has threshold one. Thus, considering an arbitrary degree-one vertex we do not choose it to be in the target set as choosing its neighbor is at least as good. Hence, this vertex does not help in activating any other vertex. This is formalized in the next data reduction rule.

Reduction Rule 2 ([22, Reduction Rule 5]) *Let $(G = (V, E), \mathrm{thr}, k)$ be an instance of TSS reduced with respect to Reduction Rule 1 and let $v \in V$ with $\mathrm{thr}(v) = \deg(v) = 1$. Then delete v from G.*

3 Parameters Related to Sparse Structures

In this section, we consider parameters that measure the sparseness of the input graph. Since trees are the most sparse connected graphs and TSS is polynomial time solvable on trees [5], parameters measuring the distance to trees are most interesting. Canonical candidates for this are the treewidth, the pathwidth, and the feedback vertex set of the input graph. Notably, a fixed-parameter algorithm of Ben-Zwi et al. [2] for the parameter treewidth tw can solve TSS in $t_{\max}^{O(\mathrm{tw})} n^{O(1)}$ time, implying fixed-parameter tractability for the three parameters mentioned above if the maximum threshold $t_{\max}$ is some constant. We extend this result by showing W[1]-hardness for them if we, instead of limiting the maximum threshold by a constant, require the thresholds to respect the majority condition. The proof even shows hardness for the combination of the feedback vertex set, the pathwidth, the vertex-deletion-distance to cograph, and the vertex-deletion-distance to interval graph. The proof is based on and further extends the W[1]-hardness reduction presented by Ben-Zwi et al. [2].

Theorem 1. Target Set Selection *with majority threshold is $W[1]$-hard even with respect to the combination of the following parameters: feedback vertex set, distance to cograph, distance to interval graph, and pathwidth.*

see Proof 1 (appendix)

Bandwidth. Another possible measure for sparseness is the bandwidth of the input. Here, our result is of more positive nature.

Theorem 2. Target Set Selection *is fixed-parameter tractable with respect to the parameter bandwidth.*

Proof. Let $(G = (V, E), \mathrm{thr}, k)$ be an instance of TSS. First, exhaustively apply Reduction Rule 1 to get a new equivalent instance $(G' = (V', E'), \mathrm{thr}', k')$. Let bw denote the bandwidth of G'. It is not hard to show that $\mathrm{thr}'(v) \leq \deg_{G'}(v) \leq 2\,\mathrm{bw}$ for all $v \in V'$. Moreover, Ben-Zwi et al. [2] gave a $(t_{\max})^{O(\mathrm{tw})}n$-time algorithm for solving TSS, where tw is the treewidth of the input graph and $t_{\max}$ is the maximum threshold value. It follows that this algorithm applied to G runs in time $(2\,\mathrm{bw})^{O(\mathrm{bw})}n$ since $\mathrm{tw} \leq 2\,\mathrm{bw}$. □

4 Parameters Related to Dense Structures

In contrast to the previous section, we now consider TSS with respect to parameters related to the denseness of the input graph. Since cliques are the most dense graphs and TSS is polynomial time solvable on cliques [22], parameters measuring the distance to cliques are most interesting. In particular, we consider the vertex deletion distance to clique and to a collection of disjoint cliques (also called cluster vertex deletion set size), and the clique cover number.

Unrestricted Thresholds. Here, we research the general TSS setting without constraints on the thresholds of the input. As hinted in the introduction, these variants are hard with respect to the denseness measures employed.

Theorem 3. Target Set Selection *is W[1]-hard with respect to the parameter "cvd number".*

see Proof 2 (appendix)

Theorem 4. Target Set Selection *is NP-hard and $W[2]$-hard with respect to the parameter "target set size" k, even on graphs with clique cover number two.*

see Proof 3 (appendix)

Restricted Thresholds. In the spirit of researching the influence of bounded thresholds to TSS, we consider the parameters distance to clique and cluster vertex deletion number (cvd number). Recall that we showed W[1]-hardness for the parameter cvd number (for unbounded thresholds) in the previous paragraph. By presenting an exponential-size problem kernel, we show that the problem becomes tractable for this parameter if the maximum threshold is a constant.

First, we show that TSS with majority thresholds or constant thresholds is fixed-parameter tractable with respect to the parameter distance ℓ to clique. We can even show fixed-parameter tractability for less restrictive threshold functions. To this end let $\mathcal{P}(V)$ be the power set of V.

Theorem 5. Target Set Selection *is fixed-parameter tractable with respect to the parameter distance ℓ to clique when the threshold function* thr *fulfills the restriction "$\mathrm{thr}(v) > g(\ell) \Rightarrow \mathrm{thr}(v) = f(N(v))$" for all vertices $v \in V$ and arbitrary functions $f : \mathcal{P}(V) \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.*

Proof. We prove the theorem by giving a fixed-parameter algorithm computing a minimum-size target set for (G, thr). To this end, we introduce some notations. Let $X \subset V$, $|X| = \ell$, denote a set of vertices such that $G[V \setminus X]$ is a clique. We define a non-standard "twins" equivalence relation $\equiv$ by

$$u \equiv v \iff (N[u] = N[v]) \wedge (\mathrm{thr}(u) = \mathrm{thr}(v)) \wedge (u \in X \iff v \in X).$$

Since the thresholds and neighborhoods of all vertices in an equivalence class Z are equal, we can denote this threshold and this neighborhood by $\mathrm{thr}(Z)$ and $N[Z]$, respectively. Let $Z_1, Z_2, \ldots, Z_s$ be a list of all nonempty equivalence classes of $\equiv$. Since $G[V \setminus X]$ is a clique, we know that for all $u, v \in V \setminus X : N[u] = N[v]$ if and only if $N_{G[X]}[u] = N_{G[X]}[v]$. Due to the condition $\mathrm{thr}(v) > g(\ell) \Rightarrow \mathrm{thr}(v) = f(N(v))$, for each subset $X' \subseteq X$, there are at most $g(\ell) + 1$ equivalence classes disjoint from X whose neighborhood in X is exactly X'. Hence, $s \leq 2^\ell(g(\ell)+1)+\ell$.

Let S be a minimum-size target set for (G, thr). With S, we can define r_i as the number of the first activation round in which all vertices of Z_i are active. More formally, $r_i := \min\{j \mid Z_i \subseteq \mathcal{A}^j_{G,\mathrm{thr}}(S)\}$. Let $r := \max\{r_i \mid 1 \leq i \leq s\}$.

In the following, we upper-bound r by s. We do this by showing that for each $1 \leq j \leq r$, there is an $1 \leq i \leq s$ such that $r_i = j$. Assume this was false, that is, there is some activation round j such that none of the equivalence classes gets activated in round j. Since $j \leq r$, there is some vertex v that gets activated in round j. Let Z_i denote the equivalence class of v. Since $j \geq 1$, we know that $|N(v) \cap \mathcal{A}^{j-1}_{G,\mathrm{thr}}(S)| \geq \mathrm{thr}(v)$. Since for each vertex $u \in Z_i$, $\mathrm{thr}(u) = \mathrm{thr}(v)$ and $N(u) = N(v)$, we conclude that $Z_i \subseteq \mathcal{A}^j_{G,\mathrm{thr}}(S)$, contradicting the assumption that $r_i \neq j$.

Now we describe our algorithm. In the first phase, we guess the correct values of r_i for all $1 \leq i \leq s$. There are at most $r^s \leq s^s$ possibilities to do so.

In the second phase of the algorithm, we use an ILP formulation to solve the problem. Each variable x_i in the ILP represents the number of vertices in the equivalence class i that are in the target set S. We use the constraints to model the activation process: For each equivalence class Z_i, the number of active neighbors in round r_i exceeds $\mathrm{thr}(Z_i)$. Two types of active neighbors are considered. First, the vertices in $N[Z_i] \cap S$. Second, the vertices in all equivalence

classes $Z_j \subseteq N[Z_i]$ that are active in round i, that is, $r_j < r_i$. More formally,

$$\begin{aligned} \text{Minimize: } & \sum_{i=1}^{s} x_i \\ \text{subject to: } & \forall 1 \leq i \leq s : \mathrm{thr}(Z_i) \leq \sum_{\substack{Z_j \subseteq N[Z_i] \\ r_j \geq r_i}} x_j + \sum_{\substack{Z_j \subseteq N[Z_i] \\ r_j < r_i}} |Z_j| \\ & \forall 1 \leq i \leq s : x_i \in \{0, 1\}. \end{aligned}$$

By the discussion above, a solution to this ILP corresponds to a minimum-size target set for (G, thr). Since the ILP formulation has s variables, a result by Lenstra [20] implies that solving it is fixed-parameter tractable with respect to s. Since at most s^s such ILPs have to be solved and $s \leq 2^\ell(g(\ell) + 1) + \ell$, fixed-parameter tractability with respect to ℓ follows. □

Clearly, Theorem 5 is a pure complexity classification result. Since the majority thresholds and constant thresholds both satisfy the restrictions required in Theorem 5, the next corollary immediately follows.

Corollary 1. Target Set Selection *with majority thresholds and constant thresholds is fixed-parameter tractable with respect to the parameter distance to clique.*

Next, we show fixed-parameter tractability for TSS with constant thresholds with respect to the parameter "cvd number". In the following, we assume that an optimal cvd set X of the input graph is given. If this is not the case, then one might instead use a simple factor-3 approximation.[8] Either way, we abbreviate $\ell := |X|$.

In this section we use the notation of "critical cliques". Here, a clique K in a graph is *critical* if all its vertices have the same closed neighborhood and K is maximal with respect to this property.

First, we present a data reduction rule allowing us to bound the number of vertices with the same open or closed neighborhood by the maximum threshold $t_{\max}$.

Reduction Rule 3 *Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of TSS that is reduced with respect to Reduction Rule 1 and let $v_1, v_2, \ldots, v_{t_{\max}+1} \in V$ be vertices such that either*

$$N(v_1) = N(v_2) = \ldots = N(v_{t_{\max}+1}) \text{ or } N[v_1] = N[v_2] = \ldots = N[v_{t_{\max}+1}].$$

Furthermore, let v_1 be the vertex with the highest threshold, that is, for all $1 \leq i \leq t_{\max} + 1$ it holds that $\mathrm{thr}(v_1) \geq \mathrm{thr}(v_i)$. Then delete v_1.

Lemma 1. *Reduction Rule 3 is correct and can be applied exhaustively in $O(n + m)$ time.*

[8] A graph is a cluster graph if and only if it contains no induced P_3, that is, an induced path of three vertices. Using this characterization, the factor 3-approximation simply deletes all vertices occurring in an induced P_3.

see Proof 4 (appendix)

In the following we assume that the input graph G is reduced with respect to Reduction Rule 1 and Reduction Rule 3. Thus, $G[V \setminus X]$ consists of disjoint cliques, each of size at most $2^\ell t_{\max}$. Hence, in order to show a problem kernel it remains to bound the number of cliques in $G[V \setminus X]$. To this end, we introduce the following notation:

Definition 1. Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of TSS, let $X \subseteq V$ be a cvd set, and let $S \subseteq V$. Let $C_1, C_2 \subseteq V$ be two clusters in $G[V \setminus X]$. We call C_1 and C_2 *equivalent with respect to* X, denoted by $C_1 \equiv_X C_2$, if there exists a bijection $f : C_1 \to C_2$ such that for every $v \in C_1$ it holds that $\mathrm{thr}(v) = \mathrm{thr}(f(v))$ and $N(v) \cap X = N(f(v)) \cap X$. Furthermore, we call C_1 and C_2 *equivalent with respect to* X *and* S, denoted by $C_1 \equiv_X^S C_2$, if the bijection f additionally fulfills $v \in S \iff f(v) \in S$ for all $v \in C_1$.

Note that $\equiv_X$ is an equivalence relation on the clusters in $G[V \setminus X]$ with at most $(t_{\max}+1)^{2^\ell t_{\max}}$ equivalence classes. To see this, observe that each equivalence class is uniquely determined by 2^ℓ (possibly empty) sequences of thresholds. One for each subset of X. Since G is reduced with respect to Reduction Rule 3, each such sequence contains between 0 and $t_{\max}$ thresholds. Since each threshold is at most $t_{\max}$, the number of equivalence classes is at most

$$\left(\sum_{i=0}^{t_{\max}} t_{\max}^i\right)^{2^\ell} \leq \left((t_{\max}+1)^{t_{\max}}\right)^{2^\ell} = (t_{\max}+1)^{2^\ell t_{\max}}.$$

In the following, our goal is to bound the number of cliques in one equivalence class in a function depending only on $t_{\max}$ and ℓ. Note that once we achieve this goal, we have a problem kernel with respect to the parameter "cvd number". The next lemma is a first step towards this goal.

Lemma 2. *Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of TSS, let $X \subseteq V$ be a cvd set for G, and let $S \subseteq V$, $|S| \leq k$, be a target set for G. Furthermore let $C_1, C_2, \ldots, C_{t_{\max}+1} \subseteq V$ be clusters in $G[V \setminus X]$ that are pairwise equivalent with respect to X and S. Then, $S \setminus C_1$ is a target set for $G[V \setminus C_1]$.*

Proof. Let $S' = S \setminus C_1$ and $G' = G[V \setminus C_1]$. We prove the lemma by contradiction: Assume that S' is not a target set for G'. Let $Y \subseteq V \setminus C_1$ be the set of vertices that are activated in G in some round i but are not activated in G' in the round i. Formally, $Y := \{v \in V \setminus C_1 \mid \exists i \geq 1 : v \in \mathcal{A}^i_{G,\mathrm{thr}}(S) \wedge v \notin \mathcal{A}^i_{G',\mathrm{thr}}(S')\}$. Since S' is not a target set for G', the set Y is not empty. In particular, Y contains all vertices in G' that are not activated by S'. Let $v \in Y$ be the vertex that is activated first in G, that is, for all $u \in Y$ it holds that $u \in \mathcal{A}^i_{G,\mathrm{thr}}(S) \Rightarrow v \in \mathcal{A}^i_{G,\mathrm{thr}}(S)$, $1 \leq i$.

Since $v \in Y$ and $Y \subseteq V \setminus C_1$, it holds that $v \notin S$. Let $i \geq 1$ be the round in which v becomes active in G, that is, $v \in \mathcal{A}^i_{G,\mathrm{thr}}(S) \setminus \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)$. Thus, $|N_G(v) \cap \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)| \geq \mathrm{thr}(v)$. Since v is in G' not activated by S', it follows that $|N_{G'}(v) \cap \mathcal{A}^{i-1}_{G',\mathrm{thr}}(S')| < \mathrm{thr}(v)$. From the selection of v it follows that $Y \cap$

$\mathcal{A}^{i-1}_{G,\text{thr}}(S) = \emptyset$. Thus, $\mathcal{A}^{i-1}_{G,\text{thr}}(S) \setminus \mathcal{A}^{i-1}_{G',\text{thr}}(S') \subseteq C_1$. Since $N_G(v) \setminus N_{G'}(v) \subseteq C_1$, it follows that $N_G(v) \cap \mathcal{A}^{i-1}_{G,\text{thr}}(S) \cap C_1 \neq \emptyset$ and $v \in X$. Let $u \in N_G(v) \cap \mathcal{A}^{i-1}_{G,\text{thr}}(S) \cap C_1$. Note that C_1 and C_j, $1 < j \leq t_{\max} + 1$, are equivalent with respect to X and S and, hence, there is a bijection f_j as described in Definition 1. Thus, it is easy to see that $u \in \mathcal{A}^{i-1}_{G,\text{thr}}(S) \Rightarrow f_j(u) \in \mathcal{A}^{i-1}_{G,\text{thr}}(S)$. Moreover, since $u \in N_G(v)$ it follows that $f_j(u) \in N_G(v)$ and, thus, $f_j(u) \in N_{G'}(v)$. Hence, $f_j(u) \in N_{G'}(v) \cap \mathcal{A}^{i-1}_{G',\text{thr}}(S')$ for all $2 \leq j \leq t_{\max} + 1$ and thus $|N_{G'}(v) \cap \mathcal{A}^{i-1}_{G',\text{thr}}(S')| \geq t_{\max}$. Hence, $\text{thr}(v) > |N_{G'}(v) \cap \mathcal{A}^{i-1}_{G',\text{thr}}(S')| \geq t_{\max}$, a contradiction. □

Since we do not know the target set S for G, two problems have to be solved in order to convert this lemma into a data reduction rule: The first problem is to find out by how much we have to decrease k, or, equivalently, how to compute $|S \cap C_1|$ in polynomial time? The second problem is that we do not know the target set S. As we show in the following, the key in overcoming these two problems is to increase the number of equivalent clusters C_j in the assumption of the lemma.

To this end, we first compute a lower and upper bound on the size of the target set for G. Let G^X be the graph that results from activating all vertices in X and applying Reduction Rule 1 exhaustively. Let $C^X_1, C^X_2, \ldots, C^X_\zeta$ denote the maximal cliques of G^X. Clearly, for each clique C^X of G^X there is a cluster C in $G[V \setminus X]$ such that $C^X \subseteq C$. Let $S^X \subseteq V$ be an optimal solution for G^X. Note that S^X can be computed in linear time [22, 26]. By construction of G^X it is clear that $|S^X|$ is a lower bound for the size of any target set for G. Furthermore, $S^X \cup X$ is a target set for G. Hence, if $k < |S^X|$ we can immediately answer no and if $k \geq |S^X| + |X| = |S^X| + \ell$ we can answer yes. Thus, we assume in the following that $|S^X| \leq k < |S^X| + \ell$. Besides these general bounds on the target set size we can also derive bounds for the number of vertices in a target set for each cluster C in $G[V \setminus X]$: If there is a (uniquely determined) clique C^X in G^X such that $C^X \subseteq C$, then set $\min(C) = |S^X \cap C^X|$. In case there is no such clique in G^X, set $\min(C) = 0$. Finally, set $\max(C) = \min\{t_{\max}, \min(C) + \ell\}$. Clearly, $\min(C)$ and $\max(C)$ are lower resp. upper bounds on the number of vertices of C that are in an optimal target set for G. Note that if two clusters C_1 and C_2 in $G[V \setminus X]$ are equivalent with respect to X, then $\min(C_1) = \min(C_2)$. Furthermore, having $\ell + 1$ clusters $C_1, \ldots, C_{\ell+1}$ in $G[V \setminus X]$ that are equivalent with respect to X, we can conclude that for any optimal target set S there is a cluster C_i, $1 \leq i \leq \ell + 1$, having exactly $\min(C_1)$ vertices in the target set, since otherwise, the solution $S^X \cup X$ for G contains fewer vertices than S. Likewise, if there are $\ell + r$ clusters $C_1, \ldots, C_{\ell+r}$ that are equivalent with respect to X, then it is clear that for any optimal target set S at least r of these clusters contain exactly $\min(C_1)$ vertices of S. Hence, increasing the number of equivalent clusters to at least $\ell + t_{\max} + 1$ solves the first problem.

We overcome the second problem by relaxing the condition for the clusters $C_1, \ldots, C_{t_{\max}} \subseteq V$ to "equivalent with respect to X" instead of "equivalent with respect to X and S" and increase the number of equivalent clusters: We can assume that, out of each cluster C, at most $\max(C) \leq t_{\max}$ vertices are in a target set. Thus, there are at most $t_{\max}^{2^\ell}$ possibilities for choosing $t_{\max}$ vertices

from a cluster to be in a target set: Choose at most $t_{\max}$ vertices with the highest threshold from each of the at most 2^ℓ critical cliques of the cluster.[9] Thus, when increasing the number of clusters that have to be equivalent with respect to X to $\ell + t_{\max}^{2^\ell}(t_{\max}+1)$ we can conclude with the pigeonhole principle that there are clusters $C_{i_1}, \ldots, C_{i_{t_{\max}+1}}$ that are equivalent with respect to X and S for any target set S and each cluster C_{i_j} contains $\min(C_{i_j})$ vertices of S. Hence, applying Lemma 2 to this set we arrive at the following reduction rule.

Reduction Rule 4 *Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of TSS that is reduced with respect to Reduction Rule 1 and let $X \subseteq V$ be a cvd set of size ℓ. Let $C_1, C_2, \ldots, C_\alpha \subset V$ be disjoint clusters in $G[V \setminus X]$ such that $\alpha = \ell + t_{\max}^{2^\ell}(t_{\max}+1)$ and for each pair C_i, C_j, $1 \leq i, j \leq \alpha$, it holds that $C_i \equiv_X C_j$. Then delete C_1 and reduce k by $\min(C_1)$.*

The correctness of the data reduction rule follows from Lemma 2 and the above discussion. As to the running time, note that Reduction Rule 4 can be exhaustively applied in $O(n^2)$ time. Since we require that the cvd set X is given, we can compute the clusters in $G[V \setminus X]$ in linear time. Then, we sort the vertices in these clusters by neighborhood and threshold. This can be done in $O(n \log(n))$ time. After this sorting the check whether two clusters are equivalent with respect to X can be done in linear time: Simply iterate over the sorted vertices and check whether the current vertices in both clusters have the same neighborhood and threshold. Overall, iterating over all clusters in $G[V \setminus X]$, determining the equivalent clusters, and deleting the respective clusters can be done in $O(n^2)$ time.

With these data reduction rules we now arrive at the following theorem.

Theorem 6. Target Set Selection *admits a problem kernel with $t_{\max}^{O(2^\ell t_{\max})}\ell$ vertices, where ℓ is the size of a cvd set and $t_{\max}$ the maximum threshold. The problem kernel can be found in $O(n^2)$ time.*

Proof. Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of TSS that is reduced with respect to Reduction Rules 1, 3, and 4. Furthermore let $X \subseteq V$ be a cvd set and let $\ell = |X|$.

Since I is reduced with respect to Reduction Rule 3, the clusters in $G[V \setminus X]$ have size at most $2^\ell t_{\max}$. Hence, there are at most $(t_{\max}+1)^{2^\ell t_{\max}}$ clusters in $G[V \setminus X]$ that are all pairwise *not* equivalent with respect to X. Furthermore, since I is reduced with respect to Reduction Rule 4, each equivalence class of $\equiv_X$ contains at most $\ell + t_{\max}^{2^\ell}(t_{\max}+1)$ clusters. Thus, the number of clusters in $G[V \setminus X]$ is bounded by $(\ell + t_{\max}^{2^\ell}(t_{\max}+1))(t_{\max}+1)^{2^\ell t_{\max}}$, each of these clusters contains at most $2^\ell t_{\max}$ vertices. Overall this gives $t_{\max}^{O(2^\ell t_{\max})}\ell$ vertices in $G[V \setminus X]$ and, thus, G contains at most $t_{\max}^{O(2^\ell t_{\max})}\ell$ vertices. The Reduction Rules 1 and 3 can both be applied exhaustively in $O(n+m)$ time and Reduction

[9] Having a set of vertices with the same closed neighborhood and the task is to choose s of them to be in a target set, it is best to choose the s vertices with the highest thresholds [22, Observation 7].

Rule 4 can be applied exhaustively in $O(n^2)$. Overall, the kernelization runs in $O(n^2)$ time. □

Clearly this problem kernel implies that TSS is fixed-parameter tractable with respect to the combined parameter $(t_{\max}, \ell)$. This leads to our next corollary considering TSS with constant thresholds.

Corollary 2. Target Set Selection *with constant thresholds is fixed-parameter tractable with respect to the parameter "cvd number".*

5 Conclusion

We showed that constant threshold values, as can be assumed in several real-world applications of TSS, can help to find efficient algorithms that exactly solve TSS. This extends previous work of Ben-Zwi et al. [2] where this observation was made for the parameter treewidth. A question left open in our work is whether or not TSS is fixed-parameter tractable with respect to the parameter cluster vertex deletion number for majority thresholds. A second open question arising from our work is whether or not TSS is fixed-parameter tractable with respect to the parameter "distance to clique" for general thresholds. Indeed, these two cases are part of the more general open question whether, in terms of computational complexity, TSS with majority thresholds is basically as hard as for general thresholds but significantly easier for constant thresholds[10]—the results we achieved in this paper may be interpreted as directing to a corresponding conjecture. Considering the practical relevance of TSS, it would be interesting to incorporate further natural parameters into the search for islands of tractability; among these we clearly have "graph diameter" (note, however, that this parameter needs to be combined with others since TSS is already hard on diameter-two graphs [22]) and "number of activation rounds" (the case of only activation round—that is, the non-dynamic setting—leads to variants of domination [14, 18, 25]; again, in order to lead to tractability results, this parameter needs to be combined with others [22]).

Bibliography

[1] J. Balogh, B. Bollobás, and R. Morris. Bootstrap percolation in high dimensions. *Combinatorics, Probability & Computing*, 19(5-6):643–692, 2010. Cited on p. 2.

[2] 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. Cited on pp. 2, 3, 4, 6, 7, 13, 16, and 17.

[3] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. 4th IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009. Cited on p. 5.

[4] 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. Cited on p. 2.

[10] Recall that majority thresholds are of particular interest in distributed computing [24].

[5] N. Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009. Cited on pp. 1, 2, 3, and 6.
[6] 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*, 2012. Online available. Cited on p. 2.
[7] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4th edition, 2010. Cited on p. 6.
[8] M. Doucha and J. Kratochvíl. Cluster vertex deletion: A parameterization between vertex cover and clique-width. In *Proc. 37th MFCS*, volume 7464 of *LNCS*, pages 348–359. Springer, 2012. Cited on p. 5.
[9] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999. Cited on pp. 5 and 19.
[10] 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*, 157:1615–1627, 2009. Cited on pp. 1 and 2.
[11] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010. Cited on p. 5.
[12] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. Cited on p. 5.
[13] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007. Cited on p. 5.
[14] J. Harant, A. Pruchnewski, and M. Voigt. On dominating sets and independent sets of graphs. *Combinatorics, Probability and Computing*, 8(6):547–553, 1999. Cited on pp. 1 and 13.
[15] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory of Computing Systems*, 47(1):196–217, 2010. Cited on p. 4.
[16] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. Cited on p. 5.
[17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th ACM KDD*, pages 137–146. ACM Press, 2003. Cited on p. 2.
[18] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Information Processing Letters*, 89(2):75–83, 2004. Cited on pp. 1 and 13.
[19] C. Komusiewicz and R. Niedermeier. New races in parameterized algorithmics. In *Proc. 37th MFCS*, volume 7464 of *LNCS*, pages 19–30. Springer, 2012. Cited on p. 4.
[20] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983. Cited on p. 9.
[21] R. M. McConnell and J. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *Proc. 5th SODA*, pages 536–545. ACM/SIAM, 1994. Cited on p. 22.
[22] A. Nichterlein, R. Niedermeier, J. Uhlmann, and M. Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 2012. Online available. Cited on pp. 2, 3, 6, 7, 11, 12, and 13.
[23] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. Cited on p. 5.
[24] D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282:231–257, 2002. Cited on pp. 2 and 13.

[25] V. Raman, S. Saurabh, and S. Srihari. Parameterized algorithms for generalized domination. In *Proc. 2nd COCOA*, volume 5165 of *LNCS*, pages 116–126. Springer, 2008. Cited on pp. 1 and 13.

[26] T. Reddy, D. Krishna, and C. Rangan. Variants of spreading messages. In *Proc. 4th WALCOM*, volume 5942 of *LNCS*, pages 240–251. Springer, 2010. Cited on pp. 2, 3, and 11.

A Appendix

A.1 Basic Reduction

Ben-Zwi et al. [2] proved W[1]-hardness for TSS with respect to the parameter treewidth. In the following, we recall their reduction, since it forms a basis for other W[1]-hardness reductions in this work. The reduction, which we will refer to as *basic reduction*, is from the W[1]-hard Multicolored Clique problem, which is defined as follows.

Multicolored Clique (MCC)
Input: An undirected graph $G = (V, E)$, an integer $k \geq 0$, and a coloring $\mathrm{col} : V \to \{1, \ldots, k\}$.
Question: Does G contain a multicolored clique of size k, that is, a vertex subset $V' \subseteq V$ with $|V'| = k$ such that for all $u, v \in V'$ it holds that $\{u, v\} \in E$ and $\mathrm{col}(u) \neq \mathrm{col}(v)$?

Let (G, col, k) be an MCC instance. An equivalent instance (G', thr, k') is constructed as follows. For each color $c \in \{1, \ldots, k\}$, create a *vertex-selection gadget* X_c consisting of a star whose leaves one-to-one correspond to vertices with color c in G. The center vertices of the star is called *guard*. For each pair of distinct colors $c_1, c_2 \in \{1, \ldots, k\}$, let $E_{\{c_1,c_2\}} \subseteq E$ be the set of all edges that connect vertices of color c_1 with vertices of color c_2 and create the following *edge-selection gadget* $X_{\{c_1,c_2\}}$. The edge-selection gadget $X_{\{c_1,c_2\}}$ consists of a star whose leaves one-to-one correspond to edges in $E_{\{c_1,c_2\}}$. The center vertex is called *guard*.

The second type of gadgets is a *validation gadget*. They use the arbitrary bijection $\mathrm{low} : V \to \{1, \ldots, n\}$ and the bijection $\mathrm{high} : V \to \{n, \ldots, 2n-1\}$ defined as $\mathrm{high}(v) = 2n - \mathrm{low}(v)$ for each $v \in V$. For each $\{c_1, c_2\}$ with $c_1, c_2 \in \{1, \ldots, k\}$, add two validation gadgets V_{c_1,c_2} and V_{c_2,c_1} each consisting of two vertices. Now, for each $\{u, v\} \in E_{\{c_1,c_2\}}$ such that $\mathrm{col}(v) = c_1$, connect the first validation gadget V_{c_1,c_2} as follows:

Let v' be the vertex in X_{c_1} corresponding to v. First, add $\mathrm{low}(v)$ vertices and connect them to v' and to the first vertex of V_{c_1,c_2}. Next, add $\mathrm{high}(v)$ vertices and connect them to v' and to the second vertex of V_{c_1,c_2}. Analogously, with $v_{\{u,v\}}$ denoting the vertex in $X_{\{c_1,c_2\}}$ corresponding to $\{u, v\}$, add $\mathrm{high}(v)$ vertices and connect them to $v_{\{u,v\}}$ and to the first vertex of V_{c_1,c_2}. Then, add $\mathrm{low}(v)$ vertices and connect them to $v_{\{u,v\}}$ and to the second vertex of V_{c_1,c_2}. The second validation gadget V_{c_2,c_1} is analogously connected to the vertex of X_{c_2} that corresponds to u and to $v_{\{u,v\}}$ in $X_{\{c_1,c_2\}}$.

We call all the vertices adjacent to vertices of a validation gadget *connection vertices*. The thresholds are set as follows: Guard vertices and connection vertices have threshold one, the two vertices in each validation gadget have threshold $2n$, and the remaining vertices in the selection gadgets have a threshold equal to their degree. Finally, $k' = k + \binom{k}{2}$. This completes the reduction.

As to the correctness: If the instance (G, col, k) is a yes-instance of MCC then the vertices chosen to be in the target set of G' refer to the multicolored

clique in G: For each vertex in the clique, the corresponding vertex in the vertex-selection gadget is in the target set. Furthermore, for each edge in the clique the corresponding vertex in the edge-selection gadget is in the target set. This target set activates the whole graph. In the reverse direction the validation gadgets play a central role: Each validation gadget connects a vertex-selection gadget with and edge-selection gadget. The vertices in the validation gadget only become activated if a vertex in the vertex-selection gadget and a vertex in the edge-selection gadget are in the target set such that the corresponding vertex and edge in G are incident. Basically this ensures that one has to choose vertices in G' into the target set that refer to a multicolored clique in G. We refer the reader to Ben-Zwi et al. [2] for more details.

A.2 Proof 1 (Theorem 1)

Proof. We modify the basic reduction to get the new, equivalent instance $(G'', \mathrm{thr}', k'')$ as follows. For each vertex v in a validation gadget, add $\deg_{G'}(v) - 4n$ vertices adjacent to v. Moreover, for each guard vertex v add $\deg_{G'}(v) - 2$ neighbors. Let X be the set of vertices added so far. Insert a new vertex u adjacent to all vertices in X and add $|X| + 2(k' + 2)$ vertices to the neighborhood of u. To complete the modification of the graph, for every vertex v in a selection gadget, attach $\deg_{G'}(v)$ neighbors. Finally, set $\mathrm{thr}'(v) = \lceil \deg_{G''}(v)/2 \rceil$ for all $v \in V(G'')$ and $k'' = k' + 1$. We claim that $(G'', \mathrm{thr}', k'')$ is equivalent to (G', thr, k').

"$\Rightarrow$": Suppose that there is a solution S' for $(G'', \mathrm{thr}', k'')$. First, observe that $u \in S'$ since, otherwise, u would not become active. Indeed, even if all the vertices in G'' plus k'' degree-one neighbors of u are activated, the vertex u will not be activated since its threshold is $|X| + k'' + 1$. Since u is in all solutions, we may consider the equivalent instance where u is removed together with all its neighbors (they all have threshold one and thus get activated by u). Moreover, for each removed vertex v, we have to decrease the threshold of the vertices in $N(v)$ by one. This operation leaves a graph with many degree-one vertices of threshold one. Applying Reduction Rule 2, we arrive at the instance (G', thr, k'). By correctness of Reduction Rule 2, the equivalence follows.

"$\Leftarrow$": Conversely, let S be a solution for (G', thr, k'). Since activating u and exhaustively applying Reduction Rule 2 results in (G', thr, k'), it is clear that $S \cup \{u\}$ is a target set for (G'', thr') of size $k' + 1$.

To complete the proof of the theorem, it is enough to observe that if we remove the vertex u, all guard vertices, and the validation gadgets (that is, $\binom{k}{2} + k + 1$ vertices), then we get stars and isolated vertices. □

A.3 Proof 2 (Theorem 3)

Proof. We modify the basic reduction (see Section 3) as follows. Make all connection vertices that have a common vertex in a (vertex- or edge-) selection gadget to a clique. The thresholds of the connection vertices are modified as follows: In each maximal clique of the connection vertices, an arbitrary ordering

of the vertices is fixed. Then the first vertex has threshold one, the second has threshold two, and the ith vertex has threshold i. See Figure 2 for a scheme of the reduction.

Note that each connection vertex is contained in exactly one maximal clique. Notice also that the following holds. Let C be such a maximal clique, V^1 (*resp.* V^1 and V^2) denote the validation gadget adjacent to C, and v the vertex adjacent to C in the vertex-selection gadget (*resp.* the edge-selection gadget). Then all connection vertices in C are activated if and only if either v is activated or all the vertices in V^1 (*resp.* V^1 and V^2) are activated.

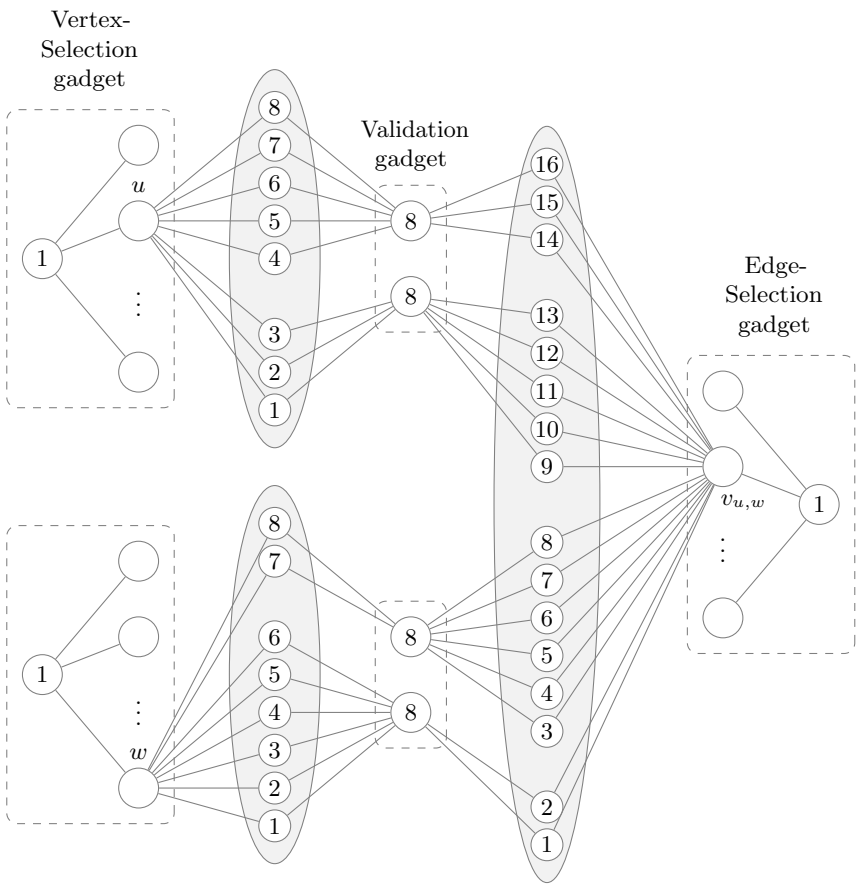

Fig. 2. Graph obtained after carrying out the modifications in the proof of Theorem 3 (with $n = 4$). The vertices inside an ellipse form a clique. The numbers in the vertices denote the thresholds. If no number is inside a vertex, the threshold is equal to the degree.

We now prove that (G, col, k) is a yes-instance of Multicolored Clique if and only if $(G', \mathrm{thr}, \binom{k}{2} + k)$ is a yes-instance of TSS.

"$\Rightarrow$": Suppose that (G, col, k) has a multicolored clique $C \subseteq V$ of size k. Then the set $S = \{v \in C\} \cup \{v_{u,v} | u, v \in C\}$ is a target set for (G', thr, k'). Indeed, in the first step of the propagation process every guard vertex is activated since they are all adjacent to a vertex in S. After $4n$ steps, all the connection vertices adjacent to a vertex in S get activated. During the next step, all $4\binom{k}{2}$ vertices in validation pairs will be activated since C is a multicolored clique of size k. From now on, it is not hard to see that the entire graph will be activated.

"$\Leftarrow$": Conversely, assume that (G', thr, k') has a target set $S \subseteq V'$ of size k. First, we may assume that S does not contain any guard vertex since they all have threshold one. Moreover, one has to pick up in the target set at least one vertex in each selection gadget to activate the guard vertex of the latter. Indeed, recall that every neighbor of a guard vertex has a threshold equal to its degree and the guard vertex is not in the target set. Thus, every target set contains at least one vertex in each selection gadget. Furthermore, since $k' = \binom{k}{2} + k$ we conclude that there is *exactly* one vertex from each selection gadget in a minimal target set. Suppose now that we select two vertices $u \in X_{c_1}$ and $v \in X_{c_2}$ together with an edge-vertex $v_{u',v'} \in X_{\{c_1,c_2\}}$ for some $c_1, c_2 \in \{1, \ldots, k\}$ such that $v_{u',v'}$ is not incident to both u and v. Without loss of generality, we may assume that $u \neq u'$. Then at least one vertex in the validation gadgets V_{c_1,c_2} and V_{c_2,c_1} will not be activated. To see this, recall that for all $w \in V'$ it holds that $\mathrm{high}(w) + \mathrm{low}(w) = 2n$ and, since $u \neq u'$, we have either $\mathrm{low}(u) + \mathrm{high}(u') < 2n$ or $\mathrm{high}(u) + \mathrm{low}(u') < 2n$. This implies, as previously discussed, that some connection vertices will not be activated, a contradiction. □

A.4 Proof 3 (Theorem 4)

Proof. We present a parameterized reduction from Hitting Set, which is $W[2]$-complete [9] with respect to the parameter "target set size".

Hitting Set (HS)
Input: A collection $\mathcal{F}$ of subsets of a finite set U and an integer $k \geq 0$.
Question: Is there a subset $U' \subseteq U$ with $U' \leq k$ such that U' contains at least one element from each subset in $\mathcal{F}$?

Given an HS-instance $(\mathcal{F}, U, k)$ consisting of a set family $\mathcal{F} = \{F_1, \ldots, F_m\}$ over a universe $U = \{u_1, \ldots, u_n\}$ and an integer $k \geq 0$, we construct a TSS-instance (G, thr, k) consisting of a graph $G = (V, E)$, a threshold function $\mathrm{thr} : V \to \mathbb{N} \cup \{0\}$, and k.

The graph $G = (V, E)$ is constructed as follows. A set of vertices V_U contains a vertex for every element $u \in U$, that is, $V_U := \{v_u \mid u \in U\}$. Analogously, a second set $W_{\mathcal{F}}$ contains a vertex for every subset, that is, $W_{\mathcal{F}} := \{w_F \mid F \in \mathcal{F}\}$. The vertices in V_U are called *element vertices* and the vertices in $W_{\mathcal{F}}$ are called

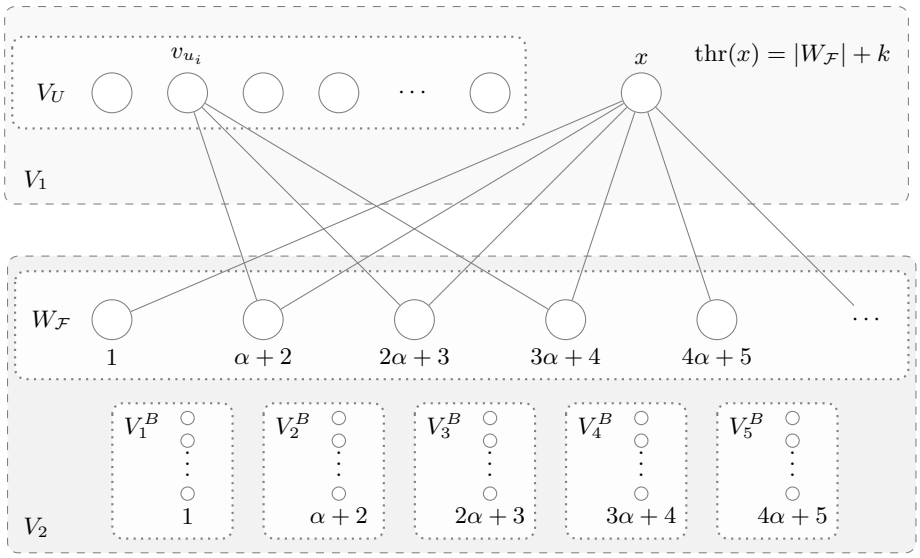

Fig. 3. A schematic picture of the constructed graph. The vertices in each box form a clique. The only way to activate all vertices with a target set of size k is to choose k vertices in V_U such that these k vertices activate all vertices in $W_{\mathcal{F}}$.

subset vertices. There is an edge between an element vertex v_u and a subset vertex w_F if and only if $u \in F$. Next, add a new vertex $x \notin (V_U \cup W_{\mathcal{F}})$ to G and connect x to all vertices in $W_{\mathcal{F}}$. Then, make $V_1 := V_U \cup \{x\}$ a clique. Add $|\mathcal{F}| - 1$ sets of vertices $V_1^B, \ldots, V_{|\mathcal{F}|-1}^B$ to the graph, each set containing $\alpha := |U| + 2$ vertices and let $V^B = \bigcup_{i=1}^{|\mathcal{F}|-1} V_i^B$. Finally, make $V_2 := W_{\mathcal{F}} \cup V_1^B \cup \ldots \cup V_{|\mathcal{F}|-1}^B$ a clique.

The thresholds are set as follows. For every subset vertex $w_{F_i} \in W_{\mathcal{F}}$, set the threshold $\mathrm{thr}(w_{F_i}) := (i-1)\alpha + i$, for every element vertex $v_u \in V_U$, set $\mathrm{thr}(v_u) := |\{F \in \mathcal{F} \mid u \in F\}| + k + 1$, and for each vertex $v \in V_i^B$, $1 \leq i \leq |\mathcal{F}| - 1$, set $\mathrm{thr}(v) := (i-1)\alpha + i$. Finally, complete the construction by setting $\mathrm{thr}(x) := |W_{\mathcal{F}}| + k$

Since V_1 and V_2 are cliques, the constructed graph G is a diameter-two graph whose vertices can be covered by two cliques, see Figure 3.

For the correctness it remains to show that $(\mathcal{F}, U, k)$ is a yes-instance of HS if and only if (G, thr, k) is a yes-instance of TSS.

"$\Rightarrow$": If $(\mathcal{F}, U, k)$ is a yes-instance, then there exists a size-k hitting set U' for $\mathcal{F}$. We show that $S := \{v_u \mid u \in U'\}$ is a size-k target set for G. Since U' is a hitting set, every vertex in $W_{\mathcal{F}}$ has at least one neighbor in S. Thus, all vertices in V_1 become active in $2|W_{\mathcal{F}}| - 1$ rounds: In the first round w_{F_1} is activated since $\mathrm{thr}(w_{F_1}) = 1$. Then in the second round, all vertices in V_1^B are activated since all these vertices also have threshold one and w_{F_1} is active. For $2 \leq i \leq |W_{\mathcal{F}}|$,

in the $(2i-1)^{\text{th}}$ round the vertex w_{F_i} is activated and in the next round all vertices in V_i^B: The neighbors of w_{F_i} that are active in round $2i-2$ are: All vertices in $V_1^B \cup \ldots \cup V_{i-1}^B$, the vertices $w_{F_1}, \ldots, w_{F_{i-1}}$ and at least one vertex in S. Since the threshold is $\text{thr}(w_{F_i}) = (i-1)\alpha + i$, the vertex w_{F_i} is activated. Then, there are $(i-1)\alpha + i$ active vertices in V_2 and, hence, all vertices in V_i^B are activated in the $2i^{\text{th}}$ round. After all vertices in V_2 are active, x is activated. Finally, in the last round all vertices in $V_U \setminus S$ are activated since for every vertex in $V_U \setminus S$ all neighbors in $W_{\mathcal{F}}$ and x have been activated.

"$\Leftarrow$": If (G, thr, k) is a yes-instance of TSS, then there is a target set S of size at most k. We first show that $S \subseteq V_U$.

Assume towards a contradiction that there is a vertex in $S \setminus V_U$. Then, $|S \cap V_U| \leq k-1$. Let ℓ denote the first round in which a vertex in $V_U \setminus S$ is activated, that is, $\ell := \min\{j \mid \mathcal{A}^j_{G,\text{thr}}(S) \cap (V_U \setminus S) \neq \emptyset\}$. Moreover, let $v_u \in \mathcal{A}^\ell_{G,\text{thr}}(S) \cap (V_U \setminus S)$. Note that, by definition of ℓ, it holds that $|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap (V_1)| \leq k$. Hence:

$$\begin{aligned}
|N_G(v_u) \cap \mathcal{A}^{\ell-1}_{G,\text{thr}}(S)| =& |N_G(v_u) \cap \mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap W_{\mathcal{F}}| \\
&+ |N_G(v_u) \cap \mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap (V_1)| \\
\leq& |\{F \in \mathcal{F} \mid u \in F\}| + k \\
<& |\{F \in \mathcal{F} \mid u \in F\}| + k + 1 = \text{thr}(v_u),
\end{aligned}$$

a contradiction. Hence, $S \subseteq V_U$.

Finally, we show that $U' := \{u \mid v_u \in S\}$ is a hitting set for $\mathcal{F}$. To this end, we show that every subset vertex w has a neighbor in S and, hence, is hit by U'. Assume towards a contradiction that there exists a vertex $w_{F_i} \in W_{\mathcal{F}}$ with $N_G(w_{F_i}) \cap S = \emptyset$. To this end let $X^i := (\bigcup_{i \leq j \leq |\mathcal{F}|-1} V_j^B) \cup (\bigcup_{i \leq j \leq |\mathcal{F}|} \{w_{F_i}\})$. Let ℓ denote the first round in which a vertex in X_i is activated, that is, $\ell := \min\{j \mid \mathcal{A}^j_{G,\text{thr}}(S) \cap X^i \neq \emptyset\}$. Hence $|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap W_{\mathcal{F}}| = |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap (W_{\mathcal{F}} \setminus X^i)| \leq i-1$ and $|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap V^B| = |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap (V^B \setminus X^i)| \leq |\bigcup_{1 \leq j < i} V_j^B| = (i-1)\alpha$. Let $v \in X^i \cap V^B$, then we have:

$$\begin{aligned}
|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap N_G(v)| &= |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap V^B| + |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap W_{\mathcal{F}}| \\
&\leq (i-1)\alpha + i - 1 \\
&< (i-1)\alpha + i \leq \text{thr}(v)
\end{aligned}$$

and, thus, $\mathcal{A}^\ell_{G,\text{thr}}(S) \cap X^i \cap V^B = \emptyset$. Now consider $v \in X^i \cap (W_{\mathcal{F}} \setminus \{w_{F_i}\})$. Observe that $|\mathcal{A}^{\ell-1}_{G,\text{thr}} \cap V_1| = |S| = k$. Thus, we have:

$$\begin{aligned}
|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap N_G(v)| &\leq |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap V^B| + |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap W_{\mathcal{F}}| + |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap V_1| \\
&\leq (i-1)\alpha + i - 1 + k \\
&< i\alpha + i + 1 \leq \text{thr}(v)
\end{aligned}$$

and, thus, $\mathcal{A}^{\ell}_{G,\text{thr}}(S) \cap X^i \cap (W_{\mathcal{F}} \setminus \{w_{F_i}\}) = \emptyset$. Finally, consider $v = w_{F_i}$:

$$\begin{aligned}|\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap N_G(v)| &\leq |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap V^B| + |\mathcal{A}^{\ell-1}_{G,\text{thr}}(S) \cap W_{\mathcal{F}}| \\ &\leq (i-1)\alpha + i - 1 \\ &< (i-1)\alpha + i = \text{thr}(v)\end{aligned}$$

and, thus, $w_{F_i} \notin \mathcal{A}^{\ell}_{G,\text{thr}}(S)$. Altogether we have $\mathcal{A}^{\ell}_{G,\text{thr}}(S) \cap X^i \cap V^B = \emptyset$, $\mathcal{A}^{\ell}_{G,\text{thr}}(S) \cap X^i \cap (W_{\mathcal{F}} \setminus \{w_{F_i}\}) = \emptyset$, and $w_{F_i} \notin \mathcal{A}^{\ell}_{G,\text{thr}}(S)$ and, hence, $\mathcal{A}^{\ell}_{G,\text{thr}}(S) \cap X^i = \emptyset$, a contradiction. □

A.5 Proof 4 (Lemma 1)

Proof. For the running time, note that computing the critical cliques of a graph can be done in linear time [21]. Thus, we first compute the critical cliques of the graph in linear time. Then we iterate over the critical cliques and if one of them has size $t_{\max} + r$, $r > 0$, then we delete the r vertices of this critical clique having the largest thresholds. This can clearly be done in linear time. Notice that a maximal set of vertices with the same open neighborhood form a critical clique in the complement graph. Hence, in a second step, we repeat the procedure with the complement graph. Then the graph is reduced with respect to Reduction Rule 3. Furthermore observe that Reduction Rule 1 is not applicable after Reduction Rule 3 was applied.

To show the correctness, we prove that the instance $(G' = (V', E'), \text{thr}, k)$ that is produced by Reduction Rule 3 is a yes-instance if and only if the input instance I is a yes-instance.

"$\Rightarrow$:" Since (G', thr, k) is a yes-instance, there exists a target set $S \subseteq V'$, $|S| \leq k$, that activates all vertices in G'. Hence, S activates all vertices of $V \setminus \{v_1\}$ in G. Since (G, thr, k) is reduced with respect to Reduction Rule 1, the vertex v_1 is activated by its neighbors. Thus, S is also a target set for (G, thr, k).

"$\Leftarrow$:" Since (G, thr, k) is a yes-instance, there exists a target set $S \subseteq V$, $|S| \leq k$, activating all vertices in G. Let $W = \{v_1, v_2, \ldots, v_{t_{\max}+1}\}$ be the vertices considered in the reduction rule. First observe that we can assume $W \setminus S \neq \emptyset$, (that is, not all vertices of W are in the target set) since otherwise, $S' = S \setminus \{v_1\}$ is also a target set: In the first activation round all vertices in $N(v_1)$ become active and, since (G, thr, k) is reduced with respect to Reduction Rule 1, it follows that v_1 is active after the second round. Thus, S' is also a target set.

Now consider the case that $v_1 \notin S$. Since for all $v_i \in W$ it holds that $\text{thr}(v_1) \geq \text{thr}(v_i)$ and v_1 is activated by its neighbors, it is clear that all vertices in $W \setminus \{v_1\}$ are active once v_1 is active. Since $|W| > t_{\max}$ this implies that all vertices in $N_G(v_1)$ become active in G' and, thus, S is a target set for G'.

Finally, consider the case that $v_1 \in S$. Let $w \in W \setminus S$ be the vertex with the highest threshold, that is, for all $v_i \in W \setminus S$ it holds that $\text{thr}(w) \geq \text{thr}(v_i)$. Observe that $S' = (S \setminus \{v_1\}) \cup \{w\}$ is a target set for G': Since S activates all vertices in W it is clear that S' activates all vertices in $W \setminus \{v_1\}$. This implies

that all vertices in $N(v_1)$ are activated by S' in G' since $|W \setminus \{v_1\}| = t_{\max}$ and all vertices in W have the same neighborhood. Thus, S' activates all vertices in G'. $\square$