Constant Thresholds Can Make Target Set Selection Tractable

**Morgan Chopin · André Nichterlein ·
Rolf Niedermeier · Mathias Weller**

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 polynomial-time approximation as well as fixed-parameter algorithms. Given an undirected graph, the task is to select a minimum number of vertices into a "target set" such that all other vertices will become active in the course of a dynamic process (which may go through several activation rounds). A vertex, 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 the existence of 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.

M. Chopin
LAMSADE, Université Paris-Dauphine, Paris, France
e-mail: chopin@lamsade.dauphine.fr

A. Nichterlein (✉) · R. Niedermeier · M. Weller
Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Berlin, Germany
e-mail: andre.nichterlein@tu-berlin.de

R. Niedermeier
e-mail: rolf.niedermeier@tu-berlin.de

M. Weller
e-mail: mathias.weller@tu-berlin.de

1 Introduction

The NP-hard TARGET SET SELECTION (TSS) problem is defined as follows: Given an undirected graph $G = (V, E)$ where each vertex $v \in V$ is assigned a positive integer threshold value $thr(v)$, 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 $thr(v)$ many of its neighbors are activated. Initially, only the vertices in S are active. TSS generalizes well-known graph problems such as DOMINATING SET with thresholds [21], VECTOR DOMINATING SET [32], k-TUPLE DOMINATING SET [25] (all these variants allow for only one "activation round"), VERTEX COVER [10] (where the threshold value equals the vertex degree), IRREVERSIBLE k-CONVERSION SET [15], r-NEIGHBOR BOOTSTRAP PERCOLATION [4] (where the threshold of each vertex is k or r, respectively), and so-called dynamic monopolies [31] (where the threshold of a vertex v with degree $\deg(v)$ is $\lceil \deg(v)/2 \rceil$—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 [10, 24] and distributed computing [31]. Indeed, since different research communities use different names to describe the problem, some work has been done independently from each other.

Since previous work has shown that TSS is computationally very hard (both in terms of approximation complexity and in terms of parameterized complexity) [6, 10, 29], 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 [10] and on cliques [29, 33], it turns hard if the treewidth is unbounded [6] (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 [29] (cliques are exactly the diameter-one graphs). This motivates the search for further parameterizations that govern the computational complexity of TSS [6, 29]. 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. Since TARGET SET SELECTION is polynomial-time solvable on forests and cliques, these parameterizations follow the "distance from triviality" approach [20]. A paragraph in Sect. 2 is dedicated to further discussing the relevance of the chosen parameters.

We are interested in the role of the allowed thresholds and one of our main conclusions is that bounding the thresholds by a constant 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. 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.

Previous Work   As mentioned before, TSS has recently received considerable interest and, together with its variants, it appears under different names and in different application contexts, making it somewhat hard to give a complete overview on the corresponding results. Thus, we focus on previous results that directly relate to our work and refrain from discussing the history of work on TSS—a more thorough review of previous work can be found in [29].

Chen [10] showed hardness of approximating TSS within a ratio of $2^{\log^{1-\epsilon}(n)}$ for any fixed $\epsilon > 0$ even for majority thresholds and thresholds at most two. For *unanimity* thresholds (that is, $\mathrm{thr}(v) = \deg(v)$ for every vertex v), TSS turns out to be equivalent to the MINIMUM VERTEX COVER problem [10] and therefore is APX-complete and fixed-parameter tractable with respect to the solution size. Very recently, Bazgan et al. [5] showed that a maximization variant of TSS (MaxTSS: select k vertices such that as many vertices as possible get activated in the end) has no parameterized approximation algorithm with respect to the parameter "target set size k"; more precisely, MaxTSS with general thresholds cannot be approximated within a factor $n^{1-\epsilon}$ in time $f(k) \cdot n^{O(1)}$ for any function f unless FPT = W[2]. Ben-Zwi et al. [6] found that TSS is W[1]-hard with respect to the parameter "treewidth" of the underlying graph. Indeed, they also showed that TSS is polynomial-time solvable on constant-treewidth graphs. However, the degree of the polynomial depends on the treewidth. 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 [29]. Moreover, polynomial-time algorithms for TSS restricted to special graph classes including chordal graphs and block-cactus graphs have been developed [8, 11, 33].

Finally, there are numerous combinatorial studies concerning the sizes of optimal target sets (upper and lower bounds) mostly with respect to special graph classes [1–3, 9, 11, 15, 34].

The role of the threshold values and threshold functions has been studied in the past. For instance, Dreyer and Roberts [15] showed NP-hardness for TSS when all vertices have the same threshold t, $t \geq 3$. Later, Chen [10] extended this result to $t = 2$. Centeno et al. [8] and Chiang et al. [11] 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. [6] who showed that TSS is W[1]-hard with respect to the treewidth of the underlying graph 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 any constant.

Our Contributions   Starting from the efficient solvability of TSS on trees and cliques [10, 29, 33], we investigate to what extent efficient algorithms can be obtained for more general graph classes. On the one hand, we consider parameters measuring tree-likeness or sparseness, thereby extending previous work [6, 29]. On the other hand, we spot several parameters measuring distance to "cliquish" graphs. In both lines, we put particular emphasis on how restricted threshold functions, in particular the majority and constant function, influence computational complexity. Notably, all our positive results for constant thresholds generalize to the case that the maximum
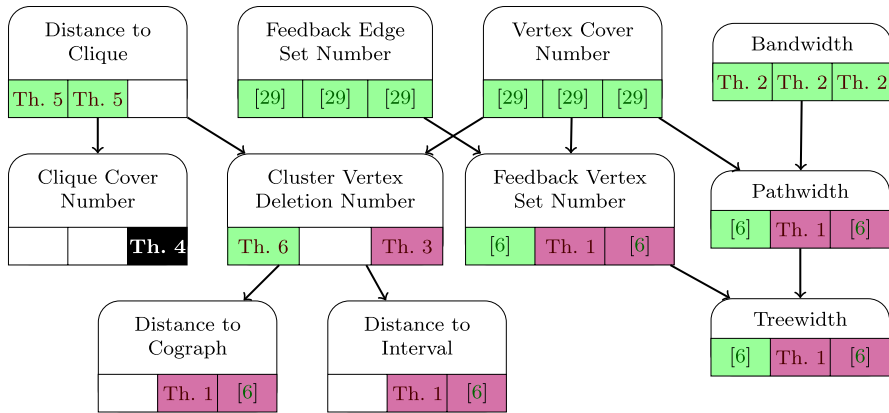
Fig. 1 Overview of the relations between structural graph parameters and our results for TARGET SET SELECTION. An arc directed from a parameter α to a parameter β means that, for all graphs, β can be upper-bounded in a linear function in α. Herein, "distance" refers to the vertex deletion distance into the specified graph class. 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 text on black background at the parameter "clique cover number" means NP-hard for constant values of this parameter, violet (dark gray in black and white version) background means W[1]-hard, green (light gray) background means FPT, and white background indicates an open question. Results marked with "Th. x" are obtained in this paper in Theorem x

threshold $t_{\max}$ is given as an additional parameter. To keep matters simple and in accordance with previous work, however, we focus on constant thresholds. Our findings, which are pictorially presented (including the relations between parameters) in Fig. 1, read as follows.

We start with the "sparse setting" (Sect. 3). 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. [6]) extend to the case of majority thresholds (Theorem 1). Conversely, the very same parameterizations lead to fixed-parameter tractability results in case of constant threshold values [6]. Further, we briefly indicate that TSS is fixed-parameter tractable for the parameter "bandwidth"[1] even in case of arbitrary threshold functions (Theorem 2).

Our main results are related to the "cliquish setting" (Sect. 4), centered around the fixed-parameter tractability of TSS with respect to the parameter "cluster vertex deletion number" (the minimum number of vertices to delete from a graph to transform it into a union of disjoint cliques [22]): TSS is W[1]-hard for general thresholds (Theorem 3) but becomes fixed-parameter tractable for constant thresholds (Theorem 6), leaving the case of majority thresholds open for future work. For the larger, also referred to as "weaker" [26], parameter "distance to clique" (the minimum number of vertices whose deletion leaves a clique), however, TSS is fixed-parameter tractable for both constant and majority thresholds (Theorem 5) whereas this is open for general

---

[1]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.

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 (Theorem 4), rendering fixed-parameter tractability very unlikely. The parameterized complexity for majority and constant thresholds is open.

Again, refer to Fig. 1 for an overview on our results in context of previous work and the respective parameter relations in the spirit of "stronger" and "weaker" parameterizations [26].

2 Preliminaries and Parameter Identification

Graph Notation   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 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 $\mathcal{A}^i_{G,\text{thr}}(S)$ with

$$\mathcal{A}^0_{G,\text{thr}}(S) := S \quad \text{and}$$

$$\mathcal{A}^{i+1}_{G,\text{thr}}(S) := \mathcal{A}^i_{G,\text{thr}}(S) \cup \big\{ v \in V \mid \big|N(v) \cap \mathcal{A}^i_{G,\text{thr}}(S)\big| \geq \text{thr}(v) \big\}.$$

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

TARGET SET SELECTION
Input: An undirected graph $G = (V, E)$, a threshold function 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 $t_{\max}(G, \text{thr}) := \max\{\text{thr}(v) \mid v \in V(G)\}$. Again, we omit (G, thr) if it is clear from the context.

A *cograph* is a graph that does not contain an induced P_4, that is, a path on four vertices. A graph $G = (V, E)$ is called *interval graph* if there exists a set of real intervals $\{I_v \mid v \in V\}$ such that $I_v \cap I_u \neq \emptyset$ if and only if $\{u, v\} \in E$ [12]. A tree decomposition of a graph $G = (V, E)$ is a pair (X, T), where $X = \{X_1, \ldots, X_n\}$ is a family of subsets of V, and T is a tree whose nodes are the subsets X_i, satisfying the following properties [12]:

1. The union of all sets X_i equals V. That is, each graph vertex is associated with at least one tree node.
2. For every edge $\{v, w\}$ in the graph, there is a subset X_i that contains both v and w.
3. If X_i and X_j both contain a vertex v, then all nodes of the tree in the (unique) path between X_i and X_j contain v as well.

A path decomposition is defined analogously with the only difference that T is required to be a path instead of a tree. The width of a tree (path) decomposition is one less than the size of its largest set $X_i \in X$. The treewidth tw (pathwidth pw) of a graph G is the minimum width among all possible tree (path) decompositions of G.

Parameterized Complexity   This is a two-dimensional framework for studying computational complexity [14, 18, 30]. 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 ℓ if it can be solved in $f(\ell) \cdot s^{O(1)}$ time, where f is a computable function only depending on ℓ. This definition also extends to *combined parameters*. Here, the parameter usually consists of a tuple of positive integers $(\ell_1, \ell_2, \ldots)$ and a parameterized problem is called fpt with respect to $(\ell_1, \ell_2, \ldots)$ if it can be solved in $f(\ell_1, \ell_2, \ldots) \cdot s^{O(1)}$ time.

A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction* [7, 19]. Here, the goal is to transform a given problem instance I with parameter ℓ in polynomial time into an equivalent instance I' with parameter $\ell' \leq \ell$ such that the size of I' is upper-bounded by some function g only depending on ℓ. If this is the case, we call I' a (problem) *kernel* of size $g(\ell)$. 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*. It is well-known that a parameterized problem is fixed-parameter tractable if and only if it has a problem kernel.

Using *parameterized reductions*, Downey and Fellows [14] developed a framework to show that problems are unlikely to be fpt. A parameterized reduction from a parameterized problem P to another parameterized problem P' is a function that, given an instance (x, ℓ), computes in $f(\ell) \cdot s^{O(1)}$ time an instance (x', ℓ') (with ℓ' only depending on ℓ) such that (x, ℓ) 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 [14, 18, 30]. Moreover, there is a whole hierarchy of classes W[t], $t \geq 1$, where, intuitively, problems become harder with growing t. In this sense, W[1]-hardness is the parameterized complexity analog of NP-hardness.

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 [16], it is natural to extract small parameters from typical properties of social networks. A widely accepted property of social networks is the so-called "small-world phenomenon", roughly stating that the diameter of social networks is usually small. Unfortunately, the diameter of the input graph turns out not to be a suitable parameter since even diameter-two graphs lead to intractability results [29].

The parameters considered in this work are derived from the observation that there are actually multiple types 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 [23] (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 consider 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 [13]. Other less restrictive parameters related to the denseness of a network such as "distance to cograph" and "distance to interval" are also considered.

In some applications, we deal with very sparse social networks, for instance networks modeling sexual contacts [16, Chap. 2, Fig. 2.7]. In these cases, parameters related to the sparseness of the input graph are interesting. Hence, 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 ("feedback edge set number" and "feedback vertex set number") as well as some graph width parameters (treewidth, pathwidth, bandwidth).

Note that all these parameters except the diameter and the "feedback edge set number" are NP-hard to compute. However, for the two parameters for which we present positive results—namely "distance to clique" (that is the size of minimum vertex cover in the complement graph) and "cluster vertex deletion number"—there exist constant-factor polynomial time approximation algorithms. Furthermore, there exist simple search tree algorithms for computing these parameters showing that they are fixed-parameter tractable with respect to their solution size [30].

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

If the threshold of a vertex exceeds its degree, then 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 [29, 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.*

In an instance that is reduced with respect to Reduction Rule 1, every degree-one vertex has threshold one. Thus, considering an arbitrary degree-one vertex, we do not select it into the target set as choosing its neighbor is at least as good. This is formalized in the next data reduction rule.

Reduction Rule 2 [29, Reduction Rule 5] *Let (G, thr, k) be an instance of TSS reduced with respect to Reduction Rule 1 and let $v \in V(G)$ 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 [10], parameters measuring the distance to trees are most interesting. Canonical candidates for this are the treewidth, the pathwidth, and the feedback

vertex set number of the input graph. Notably, if the maximum threshold t_{max} is bounded by a constant, then a fixed-parameter algorithm of Ben-Zwi et al. [6] for the parameter "treewidth tw" can solve TSS in $t_{max}^{O(tw)} \cdot n^{O(1)}$ time, implying fixed-parameter tractability for the three parameters mentioned above. Furthermore, Ben-Zwi et al. [6] proved W[1]-hardness for TSS with respect to the parameter "treewidth" when the thresholds are unbounded. We extend this result by showing W[1]-hardness for treewidth when the thresholds respect the majority condition. The proof even shows hardness for the combined parameter "feedback vertex set, pathwidth, distance to cographs, and distance to interval graphs". Finally, we show that TSS is fixed-parameter tractable when parameterized by the bandwidth. This result even holds for general thresholds.

3.1 Basic Reduction

In the following, we recall the reduction of Ben-Zwi et al. [6], 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 [17], which is defined as follows.

MULTICOLORED CLIQUE (MCC)

Input: An undirected graph $G = (V, E)$, an integer $k \geq 0$, and a coloring $col : V \rightarrow \{1, \ldots, k\}$.

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 $col(u) \neq col(v)$?

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. 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 of any star is called *guard*.

The second type of gadgets is a *validation gadget*. They use the arbitrary bijection $low : V \rightarrow \{1, \ldots, n\}$ and the bijection $high : V \rightarrow \{n, \ldots, 2n - 1\}$ defined as $high(v) := 2n - low(v)$ 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 $col(v) = c_1$, 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 $low(v)$ vertices and connect them to v' and to the first vertex of V_{c_1, c_2}. Next, add $high(v)$ vertices and connect them to v' and to the second vertex of V_{c_1, c_2}. Analogously, denoting with $e_{\{u, v\}}$ the vertex in $X_{\{c_1, c_2\}}$ corresponding to $\{u, v\}$, add $high(v)$ vertices and connect them to $e_{\{u, v\}}$ and to the first vertex of V_{c_1, c_2}. Then, add $low(v)$ vertices and connect them to $e_{\{u, v\}}$ 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 $e_{\{u, v\}}$ 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. [6] for more details.

While it is not explicitly stated by Ben-Zwi et al. [6], the presented basic reduction shows W[1]-hardness for general thresholds with respect to the combined parameter feedback vertex set, distance to cograph, distance to interval graph, and pathwidth. To see this, first observe that once we have deleted all guard vertices and validation gadgets (that is, $\binom{k}{2} + k$ vertices) in G', we get a new graph G'' which consists of stars and isolated vertices. Thus, G'' contains no cycles and is both a cograph and an interval graph implying W[1]-hardness with respect to the combined parameter feedback vertex set, distance to cograph, and distance to interval graph. Finally, the graph G' has pathwidth $O(k^2)$ which implies W[1]-hardness with respect to the pathwidth: Indeed, one can add the $O(k^2)$ deleted vertices to every node of a path decomposition of G'' of width 1 (such path decomposition exists since G'' is a collection of stars and isolated vertices and, thus, has pathwidth 1).

3.2 Extending the Basic Reduction to Majority Thresholds

In the following, we show that the basic reduction can be extended to the majority case.

Theorem 1 TARGET SET SELECTION *with majority threshold is W[1]-hard even with respect to the combined parameter feedback vertex set, distance to cograph, distance to interval graph, and pathwidth.*

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').

"⇒": Suppose that there is a solution S' for (G'', 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 Rule 2, the equivalence follows.

"⇐": 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. Therefore, using the same arguments as before, the result follows.   □

3.3 Bandwidth

Another measure for sparseness is the bandwidth of the input. Here, our result is of more positive nature: we observe that TARGET SET SELECTION is fixed-parameter tractable with respect to the bandwidth, even for general threshold functions, by using an algorithm of Ben-Zwi et al. [6].

Theorem 2 TARGET SET SELECTION *is fixed-parameter tractable with respect to the parameter "bandwidth".*

Proof Let $(G = (V, E), \text{thr}, k)$ be an instance of TSS. First, exhaustively apply Reduction Rule 1 to get a new equivalent instance $(G' = (V', E'), \text{thr}', k')$. Observe that $\text{thr}'(v) \leq \deg_{G'}(v)$ for all $v \in V'$. Let bw denote the bandwidth of G'. By the definition of bandwidth it follows that $\deg_{G'}(v) \leq 2 \cdot \text{bw}$ and, thus, $\text{thr}'(v) \leq 2 \cdot \text{bw}$ for all $v \in V'$. Moreover, Ben-Zwi et al. [6] gave a $(t_{\max})^{O(\text{tw})} \cdot n$-time algorithm for solving TSS, where tw is the treewidth of the input graph and $t_{\max}$ is the maximum threshold value. Since $\text{tw} \leq 2 \cdot \text{bw}$, this algorithm applied to G runs in $(2\text{bw})^{O(\text{bw})} \cdot n$ time.   □

4 Parameters Related to Dense Structures

In contrast to Sect. 3, 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 [29, 33], parameters measuring the distance to cliques are most interesting. In particular, we consider the vertex deletion distance to a clique and to a collection of disjoint cliques (also called "cluster vertex deletion number" or "cvd number" for short), and the clique cover number.

Starting with the case of unrestricted thresholds in Sect. 4.1, we show that TSS parameterized by the size of a minimum cluster vertex deletion (cvd) set is W[1]-hard. Furthermore, we show NP-hardness when restricting TSS to instances with clique cover number two. Then, in Sect. 4.2, we study restricted threshold functions. For constant or majority thresholds, TSS parameterized by the distance to a clique is fixed-parameter tractable. Furthermore, we show an exponential-size problem kernel for TSS with respect to the combined parameter maximum threshold value and cvd number, implying fixed-parameter tractability with respect to the cvd number on inputs with thresholds bounded by a constant.

4.1 Unrestricted Thresholds

In the following, we consider the general TSS setting without constraints on the thresholds of the input. The next two theorems state that these variants are (presumably) parameterized intractable with respect to the employed denseness measures.

Theorem 3 TARGET SET SELECTION *is W*[1]*-hard with respect to the parameter "cvd number"*.

Proof We modify the basic reduction (see Sect. 3.1) 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 Fig. 2 for a scheme of the reduction.

Note that each connection vertex is contained in exactly one maximal clique. Also 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.

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.

"⇒": Suppose that (G, col, k) has a multicolored clique $C \subseteq V$ of size k. Then the set $S := \{v \in C\} \cup \{e_{u,v} \mid u, v \in C\}$ is a target set for (G', thr, k'). Indeed, in the first step of the propagation process all guard vertices are 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.

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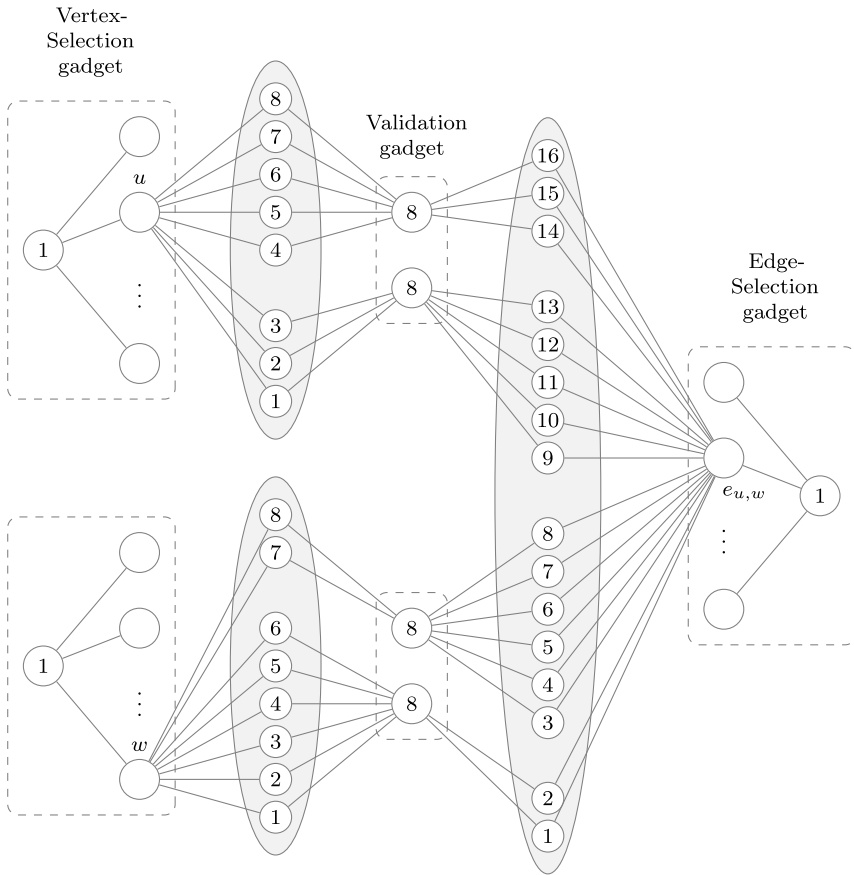
Fig. 2 Graph obtained after carrying out the modifications in the proof of Theorem 3 (with $n = 4$). The vertices inside an ellipse form a clique. The numbers in the vertices denote the thresholds. If no number is inside a vertex, the threshold is equal to the degree

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 $e_{u',v'} \in X_{\{c_1,c_2\}}$ for some $c_1, c_2 \in \{1, \ldots, k\}$ such that $e_{u',v'}$ 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 $\text{high}(w) + \text{low}(w) = 2n$ and, since $u \neq u'$, we have either $\text{low}(u) + \text{high}(u') < 2n$ or $\text{high}(u) + \text{low}(u') < 2n$. This implies, as previously discussed, that some connection vertices will not be activated, a contradiction. $\qquad \square$

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.*

Proof We present a parameterized reduction from HITTING SET, which is W[2]-complete [14] with respect to the parameter "solution size k".

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 $\text{thr} : V \rightarrow \mathbb{N} \cup \{0\}$, and k as follows.

We start with the construction of the graph G. The set V_U of vertices contains a vertex for every element $u \in U$, that is, $V_U := \{v_u \mid u \in U\}$. Analogously, the 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 *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 $V_2 := W_{\mathcal{F}} \cup V_1^B \cup \cdots \cup V_{|\mathcal{F}|-1}^B$ a clique.

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

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 Fig. 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 $\text{thr}(w_{F_1}) = 1$. 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 the following: all vertices in $V_1^B \cup \cdots \cup V_{i-1}^B$, the vertices $w_{F_1}, \ldots, w_{F_{i-1}}$, and at least one vertex in S. Since the threshold $\text{thr}(w_{F_i})$ is $(i-1)\alpha + i$, 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
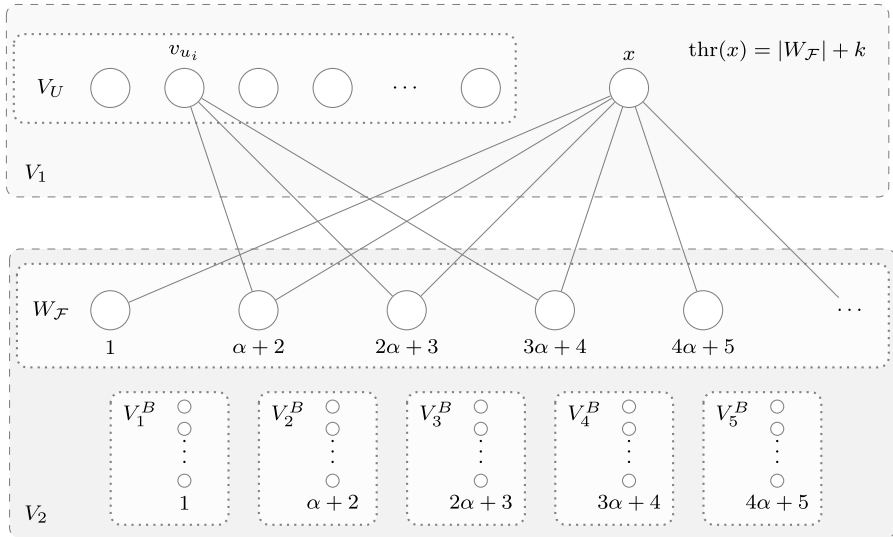
Fig. 3 A schematic picture of the constructed graph. Each of the two vertex sets V_1 (*upper box*) and V_2 (*lower box*) forms a clique. The number below a vertex denotes its threshold. The only way to activate all vertices with a target set of size k is to choose k vertices in V_U such that these k vertices activate all vertices in $W_{\mathcal{F}}$

is, $\ell := \min\{j \mid \mathcal{A}^j_{G,\mathrm{thr}}(S) \cap (V_U \setminus S) \neq \emptyset\}$. Moreover, let $v_u \in \mathcal{A}^\ell_{G,\mathrm{thr}}(S) \cap (V_U \setminus S)$. Note that, by definition of ℓ, it holds that $|\mathcal{A}^{\ell-1}_{G,\mathrm{thr}}(S) \cap (V_1)| \leq k$. Hence:

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

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

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_{F_i} 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 $N_G(w_{F_i}) \cap S = \emptyset$. Let $X^i := (\bigcup_{i \leq j \leq |\mathcal{F}|-1} V^B_j) \cup (\bigcup_{i \leq j \leq |\mathcal{F}|} \{w_{F_i}\})$. Let ℓ denote the first round in which a vertex in X^i is activated, that is, $\ell := \min\{j \mid \mathcal{A}^j_{G,\mathrm{thr}}(S) \cap X^i \neq \emptyset\}$.

Hence:

$$
\left| \mathcal{A}^{\ell-1}_{G,\mathrm{thr}}(S) \cap W_{\mathcal{F}} \right| = \left| \mathcal{A}^{\ell-1}_{G,\mathrm{thr}}(S) \cap \left(W_{\mathcal{F}} \setminus X^i \right) \right| \leq i - 1 \quad \text{and}
$$

$$
\left| \mathcal{A}^{\ell-1}_{G,\mathrm{thr}}(S) \cap V^B \right| = \left| \mathcal{A}^{\ell-1}_{G,\mathrm{thr}}(S) \cap \left(V^B \setminus X^i \right) \right| \leq \left| \bigcup_{1 \leq j < i} V^B_j \right| = (i-1)\alpha.
$$

Let $v \in X^i \cap V^B$, then we have:

$$\left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap N_G(v) \right| = \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap V^B \right| + \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap W_{\mathcal{F}} \right|$$

$$\leq (i-1)\alpha + i - 1$$

$$< (i-1)\alpha + i \leq \mathrm{thr}(v)$$

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

$$\left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap N_G(v) \right| \leq \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap V^B \right| + \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap W_{\mathcal{F}} \right| + \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap V_1 \right|$$

$$\leq (i-1)\alpha + i - 1 + k$$

$$< i\alpha + i + 1 \leq \mathrm{thr}(v)$$

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

$$\left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap N_G(v) \right| \leq \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap V^B \right| + \left| \mathcal{A}_{G,\mathrm{thr}}^{\ell-1}(S) \cap W_{\mathcal{F}} \right|$$

$$\leq (i-1)\alpha + i - 1$$

$$< (i-1)\alpha + i = \mathrm{thr}(v)$$

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

4.2 Restricted Thresholds

In the spirit of researching the influence of bounded thresholds on 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 fixed-parameter tractable with respect to 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". Indeed, 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 *on graphs with vertex set V is fixed-parameter tractable with respect to the parameter "distance ℓ to clique" if the threshold function* thr *fulfills the restriction*

$$\mathrm{thr}(v) > g(\ell) \quad \Rightarrow \quad \mathrm{thr}(v) = f\big(N(v)\big)$$

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 notation. 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

$$u \equiv v \quad \Longleftrightarrow \quad \big(N[u] = N[v]\big) \wedge \big(\text{thr}(u) = \text{thr}(v)\big) \wedge (u \in X \Longleftrightarrow v \in X).$$

Since the thresholds and neighborhoods of all vertices in an equivalence class Z are equal, we can denote this threshold and this neighborhood by $\text{thr}(Z)$ 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 $u, v \in V \setminus X$ it holds that $N[u] = N[v]$ if and only if $N_{G[X]}[u] = N_{G[X]}[v]$. Due to the condition $\text{thr}(v) > g(\ell) \Rightarrow \text{thr}(v) = f(N(v))$, 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, $r_i := \min\{j \mid Z_i \subseteq \mathcal{A}^j_{G,\text{thr}}(S)\}$. 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 $|N(v) \cap \mathcal{A}^{j-1}_{G,\text{thr}}(S)| \geq \text{thr}(v)$. Since for each vertex $u \in Z_i$, $\text{thr}(u) = \text{thr}(v)$ and $N(u) = N(v)$, we conclude that $Z_i \subseteq \mathcal{A}^j_{G,\text{thr}}(S)$, 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 constraints to model the activation process: For each equivalence class Z_i, the number of active neighbors in round r_i have to exceed $\text{thr}(Z_i)$. 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,

$$\text{Minimize:} \quad \sum_{i=1}^{s} x_i$$

$$\text{subject to:} \quad \forall 1 \leq i \leq s : \text{thr}(Z_i) \leq \sum_{\substack{Z_j \subseteq N[Z_i] \\ r_j \geq r_i}} x_j + \sum_{\substack{Z_j \subseteq N[Z_i] \\ r_j < r_i}} |Z_j|,$$

$$\forall 1 \leq i \leq s : x_i \in \{0, 1\}.$$

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 [27] 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 or 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.[2] Either way, we abbreviate $\ell := |X|$.

In this section we use the notion 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 $I := (G = (V, E), thr, k)$ 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*

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

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 $thr(v_1) \geq thr(v_i)$. Then delete v_1.

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

Proof For the running time, note that computing the critical cliques of a graph can be done in linear time [28]. 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.

---

[2]An undirected 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.

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'. □

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 these cliques contains at most $t_{\max}$ vertices for each of the 2^ℓ neighborhoods in X. 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 $I := (G = (V, E), \text{thr}, k)$ 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 $\text{thr}(v) = \text{thr}(f(v))$ 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

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

In the following, our goal is to bound the number of cliques in each 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" in case of constant thresholds. 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* $S \subseteq V$, $|S| \le k$, *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, $Y := \{v \in V \setminus C_1 \mid \exists i \ge 1 : v \in \mathcal{A}_{G,\mathrm{thr}}^i(S) \wedge v \notin \mathcal{A}_{G',\mathrm{thr}}^i(S')\}$. 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 for $1 \le i$ that $u \in \mathcal{A}_{G,\mathrm{thr}}^i(S) \Rightarrow v \in \mathcal{A}_{G,\mathrm{thr}}^i(S)$.

Since $v \in Y$ and $Y \subseteq V \setminus C_1$, it holds that $v \notin S$. Let $i \ge 1$ be the round in which v becomes active in G, that is, $v \in \mathcal{A}_{G,\mathrm{thr}}^i(S) \setminus \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S)$. Thus, $|N_G(v) \cap \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S)| \ge \mathrm{thr}(v)$. Since v is in G' not activated by S', it follows that $|N_{G'}(v) \cap \mathcal{A}_{G',\mathrm{thr}}^{i-1}(S')| < \mathrm{thr}(v)$. From the selection of v it follows that $Y \cap \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S) = \emptyset$. Thus, $\mathcal{A}_{G,\mathrm{thr}}^{i-1}(S) \setminus \mathcal{A}_{G',\mathrm{thr}}^{i-1}(S') \subseteq C_1$. Since $N_G(v) \setminus N_{G'}(v) \subseteq C_1$, it follows that $N_G(v) \cap \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S) \cap C_1 \ne \emptyset$ and $v \in X$. Let $u \in N_G(v) \cap \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S) \cap C_1$. Note that C_1 and C_j, $1 < j \le t_{\max}+1$, 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 $u \in \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S) \Rightarrow f_j(u) \in \mathcal{A}_{G,\mathrm{thr}}^{i-1}(S)$. 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, $f_j(u) \in N_{G'}(v) \cap \mathcal{A}_{G',\mathrm{thr}}^{i-1}(S')$ for all $2 \le j \le t_{\max}+1$ and thus $|N_{G'}(v) \cap \mathcal{A}_{G',\mathrm{thr}}^{i-1}(S')| \ge t_{\max}$. Hence, $\mathrm{thr}(v) > |N_{G'}(v) \cap \mathcal{A}_{G',\mathrm{thr}}^{i-1}(S')| \ge t_{\max}$, 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 precondition of Lemma 2.

To this end, we first compute a lower bound and an 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 $C_1^X, C_2^X, \ldots, C_\zeta^X$ 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 [29, 33]. 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 $k < |S^X|$, then we can immediately answer no, and if $k \geq |S^X| + |X| = |S^X| + \ell$, then 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 "equivalent with respect to X and S" for the clusters $C_1, \ldots, C_{t_{\max}} \subseteq V$ to "equivalent with respect to X" 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. 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 [29, Observation 7]. 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 $O(|X| \cdot n^2)$ 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 $O(|X| \cdot n)$ time: Simply iterate over the sorted vertices and check whether the current vertices in both clusters have the same threshold and the same neighborhood (of size at most $|X|$). Overall, iterating over all clusters in $G[V \setminus X]$, determining the equivalent clusters, and deleting the respective clusters can be done in $O(|X| \cdot n^2)$ 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 cluster vertex deletion number and* $t_{\max}$ *is the maximum threshold. The problem kernel can be found in* $O(\ell \cdot n^2)$ *time.*

Proof Let $I := (G = (V, E), \text{thr}, k)$ 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. Reduction Rules 1 and 3 can both be applied exhaustively in $O(n + m)$ time and Reduction Rule 4 can be applied exhaustively in $O(\ell \cdot n^2)$. Overall, the kernelization runs in $O(\ell \cdot n^2)$ time.   ☐

Clearly, the problem kernel of Theorem 6 implies that TSS is fixed-parameter tractable with respect to the combined parameter $(t_{\max}, \ell)$. This yields the following result for 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 naturally occur in several real-world applications of TARGET SET SELECTION (TSS), can help to find efficient algorithms that exactly solve TSS. This extends previous work of Ben-Zwi et al. [6] 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 (we showed it to be W[1]-hard for general thresholds and fixed-parameter tractable for constant 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 (it is fixed-parameter tractable for majority and constant thresholds). Indeed, these two cases are part of the more general open question whether, in terms of computational complexity, TSS is basically as hard for majority thresholds as it is for general thresholds but significantly easier for constant thresholds—the results we achieved in this paper may be interpreted as directing to a corresponding conjecture. Recall that majority thresholds are of particular interest in distributed computing [31].

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 [29]) and "number of activation rounds" (the case of only one activation round—that is, the non-dynamic setting—leads to variants of domination [21, 25, 32]; again, in order to lead to tractability results, this parameter needs to be combined with others [29]). Finally, due to applications in social networks, the identification of tractable special cases in scale-free graphs, that is, graphs with power law degree distributions, would be of particular interest.

References

1. Ackerman, E., Ben-Zwi, O., Wolfovitz, G.: Combinatorial model and bounds for target set selection. Theor. Comput. Sci. **411**(44–46), 4017–4022 (2010)
2. Adams, S.S., Troxell, D.S., Zinnen, S.L.: Dynamic monopolies and feedback vertex sets in hexagonal grids. Comput. Math. Appl. **62**(11), 4049–4057 (2011)
3. Adams, S.S., Booth, P., Troxell, D.S., Zinnen, S.L.: Modeling the spread of fault in majority-based network systems: dynamic monopolies in triangular grids. Discrete Appl. Math. **160**(10–11), 1624–1633 (2012)
4. Balogh, J., Bollobás, B., Morris, R.: Bootstrap percolation in high dimensions. Comb. Probab. Comput. **19**(5–6), 643–692 (2010)
5. Bazgan, C., Chopin, M., Nichterlein, A., Sikora, F.: Parameterized approximability of influence in social networks. In: Proc. 19th COCOON. LNCS, vol. 7936, pp. 543–554. Springer, Berlin (2013)
6. Ben-Zwi, O., Hermelin, D., Lokshtanov, D., Newman, I.: Treewidth governs the complexity of target set selection. Discrete Optim. **8**(1), 87–96 (2011)
7. Bodlaender, H.L.: Kernelization: New upper and lower bound techniques. In: Proc. 4th IWPEC. LNCS, vol. 5917, pp. 17–37. Springer, Berlin (2009)
8. Centeno, C.C., Dourado, M.C., Penso, L.D., Rautenbach, D., Szwarcfiter, J.L.: Irreversible conversion of graphs. Theor. Comput. Sci. **412**(29), 3693–3700 (2011)
9. Chang, C.-L., Lyuu, Y.-D.: Spreading messages. Theor. Comput. Sci. **410**(27–29), 2714–2724 (2009)
10. Chen, N.: On the approximability of influence in social networks. SIAM J. Discrete Math. **23**(3), 1400–1415 (2009)
11. Chiang, C.-Y., Huang, L.-H., Li, B.-J., Wu, J., Yeh, H.-G.: Some results on the target set selection problem. J. Comb. Optim. **25**(4), 702–715 (2013)
12. Diestel, R.: Graph Theory, 4th edn. Graduate Texts in Mathematics, vol. 173. Springer, Berlin (2010)
13. Doucha, M., Kratochvíl, J.: Cluster vertex deletion: a parameterization between vertex cover and clique-width. In: Proc. 37th MFCS. LNCS, vol. 7464, pp. 348–359. Springer, Berlin (2012)
14. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Berlin (1999)
15. Dreyer, P.A. Jr., Roberts, F.S.: Irreversible k-threshold processes: graph-theoretical threshold models of the spread of disease and of opinion. Discrete Appl. Math. **157**(7), 1615–1627 (2009)
16. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, Cambridge (2010)
17. Fellows, M.R., Hermelin, D., Rosamond, F.A., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. Theor. Comput. Sci. **410**(1), 53–61 (2009)

18. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
19. Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. ACM SIGACT News **38**(1), 31–45 (2007)
20. Guo, J., Hüffner, F., Niedermeier, R.: A structural view on parameterizing problems: distance from triviality. In: Proc. 1st IWPEC. LNCS, vol. 3162, pp. 162–173. Springer, Berlin (2004)
21. Harant, J., Pruchnewski, A., Voigt, M.: On dominating sets and independent sets of graphs. Comb. Probab. Comput. **8**(6), 547–553 (1999)
22. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. Theory Comput. Syst. **47**(1), 196–217 (2010)
23. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum, New York (1972)
24. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proc. 9th ACM KDD, pp. 137–146. ACM, New York (2003)
25. Klasing, R., Laforest, C.: Hardness results and approximation algorithms of k-tuple domination in graphs. Inf. Process. Lett. **89**(2), 75–83 (2004)
26. Komusiewicz, C., Niedermeier, R.: New races in parameterized algorithmics. In: Proc. 37th MFCS. LNCS, vol. 7464, pp. 19–30. Springer, Berlin (2012)
27. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
28. McConnell, R.M., Spinrad, J.: Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In: Proc. 5th SODA, pp. 536–545. ACM/SIAM, New York/Philadelphia (1994)
29. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. Soc. Netw. Anal. Mining **3**(2), 233–256 (2013)
30. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, London (2006)
31. Peleg, D.: Local majorities, coalitions and monopolies in graphs: a review. Theor. Comput. Sci. **282**(2), 231–257 (2002)
32. Raman, V., Saurabh, S., Srihari, S.: Parameterized algorithms for generalized domination. In: Proc. 2nd COCOA. LNCS, vol. 5165, pp. 116–126. Springer, Berlin (2008)
33. Reddy, T., Krishna, D., Rangan, C.: Variants of spreading messages. In: Proc. 4th WALCOM. LNCS, vol. 5942, pp. 240–251. Springer, Berlin (2010)
34. Reichman, D.: New bounds for contagious sets. Discrete Math. **312**(10), 1812–1814 (2012)