Noname manuscript No.
(will be inserted by the editor)

On Tractable Cases of Target Set Selection

André Nichterlein · Rolf Niedermeier · Johannes Uhlmann · Mathias Weller

Abstract We study the NP-hard Target Set Selection (TSS) problem occurring in social network analysis. Roughly speaking, given a graph where each vertex is associated with a threshold, in TSS the task is to select a minimum-size "target set" such that all vertices of the graph get activated. Activation is a dynamic process. First, only the vertices in the target set are active. Then, a vertex becomes active if the number of its active neighbors exceeds its threshold, and so on. TSS models the spread of information, infections, and influence in networks.

Complementing results on its polynomial-time approximability and extending results for its restriction to trees and bounded treewidth graphs, we classify the influence of the parameters "diameter", "cluster editing number", "vertex cover number", and "feedback edge set number" of the underlying graph on the problem's computational complexity, revealing both tractable and intractable cases. For instance, even for diameter-two split graphs TSS remains $W[2]$-hard with respect to the parameter "size of the target set". TSS can be efficiently solved on graphs with small feedback edge set number and also turns out to be fixed-parameter tractable when parameterized by the vertex cover number. Both results contrast known parameterized intractability results for the parameter "treewidth". While these tractability results are relevant for sparse networks, we also show efficient fixed-parameter algorithms for the parameter "cluster editing number", yielding tractability for certain dense networks.

Keywords Dynamic Monopolies, Influence Spreading, Spread of Information, Viral Marketing, Algorithms and Complexity, Parameterized Computational Complexity, Fixed-Parameter-Tractability, Data Reduction

This work is supported by the DFG, research projects PABI, NI 369/7, and DARE, NI 369/11. An extended abstract appeared in the *Proceedings of the 21st International Symposium on Algorithms and Computation (ISAAC '10)*, Part I, volume 6506 of *LNCS*, pages 378-389, Springer 2010.

Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Berlin, Germany.
{andre.nichterlein, rolf.niedermeier, mathias.weller}@tu-berlin.de,
johannes.uhlmann@campus.tu-berlin.de

1 Introduction

In the NP-complete graph problem Target Set Selection (TSS) one is given an undirected graph $G = (V, E)$, a threshold function $\text{thr} : V \to \mathbb{N} \cup \{0\}$, and an integer $k \geq 0$, and the question is whether there is a "target set" $S \subseteq V$ for G with $|S| \leq k$ activating all vertices in V, that is, for every $v \in V \setminus S$ eventually at least $\text{thr}(v)$ of v's neighbors are active. *Activation* is a dynamic process, where initially only the vertices in S are active and the remaining vertices may become active step by step during several rounds (see Section 2 for a formal definition). Roughly speaking, TSS offers a simple model to study the spread of influence, infections, or information in social networks. While this concept was previously studied under the name "dynamic monopolies" (Peleg, 2002), Kempe et al (2003) referred to it as influence maximization with a linear threshold model.

One application of the influence spreading model is *viral marketing*. A good example for an effective viral marketing strategy is the e-mail service Hotmail: Every mail sent via a Hotmail account contained some advertisement for the service. The result was impressive: Launched July 1996 with a budget of $50,000 for advertisement, Hotmail grew to 12 million users in 18 months (Jurvetson, 2000). The name viral marketing expresses that this word-of-mouth advertisement can spread like an epidemic. See also Leskovec et al (2007a) for a study of a recommendation network or Leskovec et al (2007b) for a study of information or influence diffusion in blog graphs. The results achieved with viral marketing advertisement strategies can be amazing, however, it is difficult to achieve such a success (Leskovec et al, 2007a; Watts et al, 2007). Apart from viral marketing, TSS has a number of further applications. For example, Dreyer and Roberts (2009) described a scenario of an application in bioterrorism defense. Variants of TSS with special threshold functions are applicable in the field of fault-tolerant distributed computing (Peleg, 2002). Here, the influence of faulty processing units is limited by taking local majority votes of redundant units. The special threshold function used in this scenario allows modeling of a multitude of additional problems in system-level diagnosis (Pelc, 1996), distributed database management (Davidson et al, 1985), quorum systems (Garcia-Molina and Barbará, 1985) and fault-local mending (Kutten and Peleg, 1999).

It is important to note that, with our definition of TSS, we study the case where the goal is to activate *all* vertices in the network. In application contexts, such as viral marketing and bioterrorism defense, it might be more natural to only seek to *maximize* the number of activated vertices with a given number of initially active vertices (which are to be determined). However, activating all vertices as we do in our TSS model seems particularly relevant in political scenarios or in strategies to convince people of the relevance of a vaccination program. For instance, in the European Union matters related to trade or foreign and security policy often have to be decided on a 100% approval by all member states. So "complete activation" is necessary in these scenarios. Furthermore, by definition, a dynamic monopoly activates all vertices in a given graph, implying that applications revolving around dynamic monopolies usually require activation of every vertex.

Importantly, in our definition of TSS once a vertex becomes active it never falls back to non-active state. Non-monotone versions of TSS where active vertices may become non-active also make sense in several applications. However, the non-monotone as well as the non-complete activation version of TSS are both

generalizations of our model and therefore presumably harder in terms of computational complexity. Ben-Zwi et al (2011) showed hardness results for these more general TSS models. In this work we focus on the TSS model where all vertices are to be activated and active vertices stay active.

We extend previous work (Ben-Zwi et al, 2011; Chen, 2009; Kempe et al, 2003) by studying several special cases of TSS. In particular, we investigate TSS on graphs that have characteristics of social networks, including the "small-world" property (Newman, 2003, 2010). We remark that there are many further models referring to the spread of influence in social networks, for instance see recent work of Agarwal et al (2011), Raeder and Chawla (2011), and Zhang et al (2012).

Previous work. Domingos and Richardson (2001, 2002) introduced a probabilistic version of TSS, studying it from the viewpoint of viral marketing and solving it heuristically. Next, Kempe et al (2003) formulated the problem using a threshold model (as we use it now), showed its NP-completeness, and presented a polynomial-time constant-factor approximation algorithm for a maximization variant. In this maximization variant the goal is not to activate all vertices but as many as possible with a given upper bound on the target set size. Next, Chen (2009) showed the APX-hardness of minimizing the size of the target set S for the complete activation version, which holds even in case of some restricted threshold functions. He also provided a linear-time algorithm for trees. Ben-Zwi et al (2011) generalized Chen's result for trees by showing that TSS is polynomial-time solvable for graphs of constant treewidth; however, the degree of the polynomial of the running time depends on the treewidth. In other words, they showed that TSS is in the parameterized complexity class XP when parameterized by treewidth. They also proved that there is no $|V|^{o(\sqrt{\omega})}$ time algorithm (ω denoting the treewidth) unless some unexpected complexity-theoretic collapse occurs. In particular, Ben-Zwi et al (2011) showed that TSS is W[1]-hard[1] with respect to the treewidth ω and, hence, there is little hope for getting ω out of the exponent of the polynomial, that is, little hope for fixed-parameter tractability of TSS for parameter treewidth. There are numerous investigations on TSS variants: When the number of activation rounds is limited to one, we arrive at a Dominating Set[2] variant with thresholds (Harant et al, 1999). Other studies include special threshold functions: Abrahamson et al (1995) studied the t-Threshold-Starting Set problem which is the same as TSS transferred to the setting of directed graphs where all vertices have fixed threshold t. In the context of parameterized complexity they showed its W[P]-completeness with respect to the parameter target set size. Dreyer and Roberts (2009) showed NP-hardness for TSS when all vertices have a threshold of t, $t \geq 3$, and they proved lower bounds for the size of the target set on several special graph classes. Ackerman et al (2010) obtained lower and upper bounds for the size of the target set on directed graphs when the threshold of each vertex is half its degree (called "majority thresholds"). These majority thresholds are well

[1] Informally speaking, W[1]-hardness means that there is no hope for fixed-parameter tractability with respect to the corresponding parameter; we refer to Section 2 for the formal definitions.

[2] Given an undirected graph $G = (V, E)$ and a positive integer h, the Dominating Set problem asks whether there is a vertex subset $V' \subseteq V$ of size at most h such that for every $v \in V$ it holds that $v \in V'$ or v is adjacent to a vertex in V'.

Table 1 Summary of our results. The number of vertices is denoted by n and the number of edges by m.

parameter	result
treewidth	W[1]-hard (Ben-Zwi et al, 2011)
feedback vertex set	W[1]-hard (Ben-Zwi et al, 2011)
diameter	NP-hard for diameter two (Section 3)
target set size k	W[2]-hard on split graphs (Section 3)
cluster edge deletion number ξ	FPT: running time $O(4^{\xi} \cdot m + n^3)$ (Section 4)
cluster editing number ζ	FPT: running time $O(16^{\zeta} \cdot m + n^3)$ (Section 4)
vertex cover number τ	FPT: running time $O(2^{(2^{\tau}+1)\cdot\tau} \cdot m)$ (Section 5) kernel with $O(2^{\tau})$ vertices for constant thresholds
feedback edge set number f	FPT: running time $O(4^{f} \cdot f + m)$ kernel of size $O(f)$ (Section 6)

studied in context of majority voting systems, especially in distributed computing and resource allocation. We refer to Peleg (2002) for a survey on target sets for graphs with majority thresholds, called "dynamic monopolies" ("dynamos") in the survey. Here, the activation process is called "repetitive polling process" or "majority process". Again, the studies focused on lower bounds for the size of the target set. Finally, for a general account we refer to the book of Easley and Kleinberg (2010) covering the mentioned applications (viral marketing, influence spreading, majority voting systems) and mathematical cascading models (such as TSS). Recently, the following problem related to TSS received increasing attention (Wang and Moeller, 2002; MacGillivray and Wang, 2003), in particular, from a parameterized perspective (Bazgan et al, 2011; Cygan et al, 2012): Given a graph and a specific activated vertex, the task is to confine the activation process to a minimum number of vertices by making a small number of vertices "immune" in each round.

Our contributions. Roughly speaking, taking an approach in the spirit of multivariate algorithmics (Fellows, 2009; Niedermeier, 2010), we shed more light on TSS's potential islands of tractability. We also discuss (see the concluding section) the large potential for future algorithmic research on TSS.

Social networks often have special characteristics, including the "small-world" property, a power-law degree distribution, and some kind of community structure (Girvan and Newman, 2002; Leskovec and Horvitz, 2008; Leskovec et al, 2007b; Milgram, 1967; Newman, 2003, 2010; Newman and Park, 2003). This makes it desirable to better understand the computational complexity of TSS when restricted to such special input graphs. Since the mentioned characteristics are of a rather "statistical nature", we do not directly address them, but, to some extent, replace them by some combinatorial graph parameters that might be typically small in certain social networks. For instance, the "small-world" phenomenon directly relates to graphs of small diameter as we study in this work. More specifically, motivated by the mostly negative (or impractical) algorithmic results of Ben-Zwi et al (2011), we study further natural parameterizations characteristics of social networks. We obtain both hardness and tractability results. Table 1 summarizes our findings.

Our work is organized as follows. In Section 3, we focus on graphs with small diameter because the diameter of real-world social networks is often small ("small-

world" property) (Girvan and Newman, 2002; Newman, 2003, 2010; Newman and Park, 2003). Since TSS resembles a dynamic variant of the well-known Dominating Set problem, it may come as no surprise that the W[2]-hardness of Dominating Set with respect to the solution size on graphs with diameter two (Downey et al, 2008) carries over to TSS. We show that this is indeed the case. In contrast, we also observe that TSS can be solved in linear time for diameter-one graphs (that is, cliques). In addition, we show that $W[2]$-hardness persists on bipartite graphs with diameter four, even if all vertices have threshold two (note that threshold one is trivial) or the thresholds respect the "majority" condition (that is, for each vertex not in the target set S at least half of its neighbors need to be active).

The algorithmic main part of the paper considers three different parameterizations of TSS exploiting the following structural graph parameters.

In Section 4, we look at the "cluster editing number" ζ of the input graph, denoting the minimum number of edges whose addition or deletion transforms the graph into a disjoint union of cliques. This parameter is small for "highly clustered" input graphs. Although the cluster editing number may be large in general, we expect ζ to be small for some applications for which the corresponding social networks exhibit a cluster graph-like structure.[3] A smaller parameter exploiting this feature would be the "cluster vertex deletion number" of the input, that is, the minimum number of vertices whose deletion transforms the graph into a disjoint union of cliques. However, in ongoing work, we found indications that Target Set Selection is most likely not fixed-parameter tractable with respect to this stronger parameter. Here, we show that TSS can be solved in $O(16^{\zeta} \cdot m + n^3)$ time and provide a polynomial-time data reduction that yields a polynomial-size problem kernel with respect to the combination of the parameters ζ and the maximum threshold value $t_{\max}$.

In Section 5, following the spirit of previous work considering graph layout and coloring problems (Bodlaender et al, 2011a; Fellows et al, 2008; Fiala et al, 2011), we study the parameter "vertex cover number" τ of the input graph.[4] This parameter imposes a stronger restriction than treewidth (measuring the tree-likeness of a graph), that is, the treewidth is always upper-bounded by the vertex cover number. Note that TSS is W[1]-hard with respect to the parameter treewidth (Ben-Zwi et al, 2011). Using the stronger restriction of the parameter "vertex cover number", we prove that TSS is fixed-parameter tractable with respect to τ. In addition, for maximal threshold $t_{\max}$ we show a problem kernel consisting of $O(t_{\max} \cdot 2^{\tau})$ vertices and prove that there is little hope for a polynomial-size problem kernel for TSS parameterized by the vertex cover number even when using majority thresholds.

Finally, in Section 6, as a third and (other than the previous parameters) "easy-to-compute" parameter also measuring tree-likeness, we study the feedback edge set number f (the minimum number of edges to delete to make the input graph acyclic). Graphs with small feedback edge set number are "almost trees"; such social networks occur in the context of sexually transmitted infections (Potterat et al,

[3] For example, this is plausible for networks of teams (say in sports or work teams) where vertices are persons and there is an edge between two vertices if the intersection of their team membership durations is sufficiently large.

[4] Given an undirected graph $G = (V, E)$ and a positive integer h, the Vertex Cover problem asks whether there is a vertex subset $V' \subseteq V$ of size at most h such that each edge in E has at least one endpoint in V'.

2002) and extremism propagation (Franks et al, 2008). We develop polynomial-time data reduction rules that yield a linear-size problem kernel with respect to the parameter f. Moreover, we show that TSS can be solved in $O(4^f \cdot f + m)$ time, which again generalizes the linear-time algorithm of Chen (2009) for trees.

In general, we try to state hardness results in their most special form, since for example, hardness on bipartite graphs implies hardness on general graphs as well. Likewise, we present positive results in their most general form.

Parameter values in real-world networks. It is known that the diameter of social networks is usually small (Newman, 2010). However, we show in Section 3 that TSS is still NP-hard on graphs with constant diameter. On the positive side, we show fixed-parameter tractability for the parameters "cluster editing number", "vertex cover number", and "feedback edge set number". In the following we provide some evidence that there are real-world social networks where one or more of these parameters are significantly smaller than the number of vertices.

We analyzed the size of the mentioned parameters in coauthor networks derived from the DBLP dataset,[5] two networks showing the relation between PhD candidates and their advisors (Johnson, 1984; De Nooy et al, 2011), and one network showing the ownership between companies (Norlen et al, 2002).[6] Although the last three networks are directed, we considered the underlying undirected network. Note that we used simple greedy heuristics to compute the sizes of the vertex cover and the cluster editing set. Thus, the optimal values may actually be smaller.

We generated graphs from the DBLP dataset by using the following two mechanisms: First, we considered researchers from one scientific community in the network. This was achieved by considering all papers that are published in the following conferences: KDD, ICDE, CIKM, DMKD, and SDM (the five top-rated data mining conferences according to `http://academic.research.microsoft.com/`). A second way to select researchers of one scientific community is to consider the graph that is induced by influential researchers from the community (we chose fifty researchers publishing on algorithms and complexity) and their coauthors. We added an edge between two authors if they worked together on at least a specified number of publications. The reason for this thresholding is to work out the "strong ties" (Easley and Kleinberg, 2010) of the collaboration network. In the resulting graphs, we removed all isolated vertices.

Table 2 shows the parameter values for the described graphs. For the graphs generated from DBLP, the parameters "feedback edge set number" and "vertex cover number" are fairly small when filtering for the strong ties in the network. Due to the network type, it is not surprising that the feedback edge set number is small in the the last three networks. This indicates that our linear-size problem kernel and also the search tree algorithm for the parameter "feedback edge set number" are good candidates for further experimental studies. Note that our fixed-parameter tractability result for the parameter "vertex cover number" is of pure classification nature and, hence, exploiting this parameter experimentally needs further theoretical improvements. Furthermore, the cluster editing size is not that

[5] The dataset and a corresponding documentation are available online (`http://dblp.uni-trier.de/xml/`).

[6] These three networks are available in the Pajek Dataset of Vladimir Batagelj and Andrej Mrvar (2006) (`http://vlado.fmf.uni-lj.si/pub/networks/data/`).

Table 2 Parameter values for certain social networks. For the networks "DBLP conference induced", publications in the five mentioned data mining conferences are considered. For the networks "DBLP author induced", influential researchers and their coauthors are considered. The column labeled "#p" gives the number of joint publications needed for an edge to exist between two coauthors; isolated vertices are removed. "CSphd.net" and "PhD.paj" refer to the genealogy networks with the relation between advisors and students. Finally, "eva.net" reflects the ownership relation of companies.

Graph	#p	n	m	VC-size	FES-size	CE-size
DBLP conference induced	1	11133	27255	6520	16942	18586
	5	322	274	142	42	119
	10	42	27	16	0	11
DBLP author induced	1	2678	10182	1419	7505	8412
	5	355	440	85	100	358
	10	102	90	31	8	58
CSphd.net		1882	1740	555	26	1316
PhD.paj		1025	1043	263	19	326
eva.net		7253	6713	1216	183	1351

small in the given social networks, implying that our approach for the parameter "cluster editing number" is not very promising here.

2 Problem Statement and Preliminaries

In this section, we provide basic notation used throughout this work and give a formal definition of Target Set Selection. Then, we give a short introduction to parameterized algorithmics and problem kernelization.

Basic graph notation. Let $G = (V, E)$ be a graph and let $n := |V|$ and $m := |E|$ throughout this work. The *(open) neighborhood* of a vertex $v \in V$ in G is $N_G(v) := \{u : \{u, v\} \in E\}$ and the *degree* of v in G is $\deg_G(v) := |N_G(v)|$. The *closed neighborhood* of a vertex $v \in V$ in G is $N_G[v] := N_G(v) \cup \{v\}$. Moreover, for $V' \subseteq V$ let $N_G(V') := \bigcup_{v \in V'} N_G(v) \setminus V'$ and $N_G[V'] := \bigcup_{v \in V'} N_G[v]$. If G is clear from context, we just write $N(v)$, $N[v]$ and $\deg(v)$. We sometimes write $G - x$ as an abbreviation for the graph that results from G by deleting x, where x may be a vertex, an edge, a vertex set, or an edge set. The *diameter* of a graph G is the maximum length of a shortest path between two different vertices in G. A *split graph* is a graph in which the vertices can be partitioned into a clique and an independent set (Brandstädt et al, 1999).

Problem statement. Let $G = (V, E)$ be an undirected graph and let $\text{thr} : V \to \mathbb{N} \cup \{0\}$ be a threshold function. The definition of Target Set Selection is based on the notion of "activation". Let $S \subseteq V$. Informally speaking, a vertex $v \in V$ gets activated by S in the i^{th} round if at least $\text{thr}(v)$ of its neighbors are active after the previous round (where S are the vertices active in the 0^{th} round). Formally, for a vertex set S, let $\mathcal{A}^i_{G,\text{thr}}(S)$ denote the set of vertices of G that are *activated by S in the i^{th} round*, with

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

For $S \subseteq V$, the unique positive integer r with $\mathcal{A}^{r-1}_{G,\mathrm{thr}}(S) \neq \mathcal{A}^{r}_{G,\mathrm{thr}}(S) = \mathcal{A}^{r+1}_{G,\mathrm{thr}}(S)$ is called the *number $r_G(S)$ of activation rounds*. It is easy to see that $r_G(S) \leq |V(G)|$ for all graphs G. Furthermore, we call $\mathcal{A}_{G,\mathrm{thr}}(S) := \mathcal{A}^{r_G(S)}_{G,\mathrm{thr}}(S)$ the set of vertices that are *activated by S*. If $\mathcal{A}_{G,\mathrm{thr}}(S) = V$, then S is called a *target set for G*. Target Set Selection is the problem to find an optimal (that is a minimum-cardinality) target set. Its decision version is defined as follows.

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 that contains at most k vertices?

All our results are tailored for the decision version of TSS, indicated by instances (G, thr, k). Some statements, however, are made regarding the optimization variant of TSS, where the task is to find a minimum target set for the input graph G. Instances of the optimization variant are of the form (G, thr).

We also investigate the influence of the threshold function to the computational complexity of TSS.

Thresholds. Apart from allowing arbitrary threshold functions, we consider two further types of threshold functions in this work.

1. *Constant thresholds:* All vertices have the same threshold.
2. *Degree-dependent thresholds:* The threshold of a vertex v depends on $\deg(v)$. In this context, note that if $\mathrm{thr}(v) = \deg(v)$ for all vertices v, then TSS is identical to the Vertex Cover problem (Chen, 2009). In this work, we particularly consider the "*majority threshold* function", defined as $\mathrm{thr}(v) := \lceil \deg(v)/2 \rceil$ (Chen, 2009).

In our work, thresholds of vertices are sometimes decreased. However, thresholds can never be smaller than zero; further decreasing a threshold of zero has no effect.

Since solving instances of TSS with maximal threshold one is trivial (just select an arbitrary vertex in each connected component), we assume that each connected component of an input graph contains a vertex with threshold at least two.

Parameterized complexity. This is a two-dimensional framework for studying computational complexity (Downey and Fellows, 1999; Flum and Grohe, 2006; Niedermeier, 2006). One dimension is the input size n, 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 $f(k) \cdot n^{O(1)}$ 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* (Bodlaender, 2009; Guo and Niedermeier, 2007).[7] 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

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

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*.

Downey and Fellows (1999) 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 $f(k) \cdot n^{O(1)}$ 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 (Downey and Fellows, 1999; Flum and Grohe, 2006; Niedermeier, 2006). Moreover, there is a whole hierarchy of classes $W[t]$, $t \geq 1$, where, intuitively, problems become harder with growing t.

First observations on TSS. We continue with simple observations and results which are meant for (multiple) later reference throughout the paper rather than for sequential reading now. These observations are straightforward and do not need proofs.

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.

Observation 1 (Stubborn Vertices) *Let $G = (V, E)$ be a graph and let $v \in V$. If $\mathrm{thr}(v) > \deg(v)$, then v is contained in all target sets for G. If $\mathrm{thr}(v) = 0$, then v is not contained in any optimal target set for G.*

In the following, we define the influence of "activating" a vertex subset on the input graph. If a vertex v is active, then we can decrease the threshold of its neighbors by one and delete v. Combining this with Observation 1 allows us to use the following efficient data reduction rule on all input graphs.

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.*

Considering that changes propagate over each edge at most once, it is not hard to see that Reduction Rule 1 can be applied exhaustively in linear time. With Reduction Rule 1, we can define the graph that remains after an activation process.

Definition 1 Let $(G = (V, E), \mathrm{thr}, k)$ be an instance of TSS, let $S \subseteq V$, and let $\mathrm{thr}_S : V \to \mathbb{N} \cup \{0\}$ with $\mathrm{thr}_S(v) := \infty$ for all $v \in S$ and $\mathrm{thr}_S(v) := \mathrm{thr}(v)$ for all $v \in V \setminus S$. Then, we call the instance (G', thr', k') that results from exhaustively applying Reduction Rule 1 to (G, thr_S, k) the *S-reduced instance* of (G, thr, k).

Observe that reducing an instance with respect to a target set S results in the empty graph.

Clearly, instead of adding a threshold-one vertex to the target set, it is at least as good to add one of its neighbors. Thus, since we assume that there is at least one vertex with threshold at least two in each connected component, there is a target set containing only vertices with threshold at least two.

Observation 2 (Lemming Vertices) *Let $G = (V, E)$ be a connected graph reduced with respect to Reduction Rule 1. Then there is an optimal target set for G not containing vertices with threshold one.*

The next observation is helpful for modifying solutions by means of vertex exchange.

Observation 3 *Let S denote a target set for $(G = (V, E), \mathrm{thr})$ and let $S' \subseteq V$. If there are two integers i, j with $\mathcal{A}^i_{G,\mathrm{thr}}(S) \subseteq \mathcal{A}^j_{G,\mathrm{thr}}(S')$, then S' is a target set for G.*

If we add a new vertex with reasonable threshold to a graph and make it adjacent to all other vertices, then a target set for the original graph is a target set for the new graph, since the new vertex gets activated if all other vertices are active (at the latest).

Observation 4 (Vertex Addition) *Let $G = (V, E)$ be a graph and let $S \subseteq V$ be a target set for G. Let G' denote the graph that results from adding a vertex v with $\mathrm{thr}(v) \leq |V|$ to G and connecting it to $\mathrm{thr}(v)$ arbitrary vertices of G. Then S is also a target set for G'. Conversely, if S' is a target set for G' such that $v \notin S'$, then S' is also a target set for G.*

For a vertex subset $V' \subseteq V$ it makes no difference for the V'-reduced instance of (G, thr, k) whether we activate all vertices in V' at once or if the vertices of V' are activated one at a time in an arbitrary order.

Observation 5 (Chain Reduction) *Let I be an instance of* TSS *with vertex set V, let $S_1, S_2 \subseteq V$ with $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$, and let I_1 be the S_1-reduced instance of I. Then, the S_2-reduced instance of I_1 is identical to the S-reduced instance of I.*

With Observation 2 it is easy to verify that "subdividing" an edge $\{u, v\}$, that is, replacing $\{u, v\}$ with a new threshold-one vertex that is adjacent exactly to u and v, leads to an equivalent instance.

Observation 6 (Subdivision) *Let G be a graph and let $\{u, v\}$ be an edge of G. Let G' denote the graph that results from deleting the edge $\{u, v\}$ and inserting a vertex x with $\mathrm{thr}(x) = 1$ and the edges $\{x, u\}$ and $\{x, v\}$. Then, G has a target set of size k if and only if G' has a target set of size k.*

Non-adjacent vertices with the same open neighborhood are interchangeable in any target set.

Definition 2 Let $G = (V, E)$ be a graph. Two non-adjacent vertices with the same open neighborhood are called *twins*. A set V' of vertices is called *critical independent set* if each two distinct vertices from V' are twins and V' is maximal with respect to this property.

Observation 7 (Twin Exchange) *Let $G = (V, E)$ denote a graph and let $u, v \in V$ denote two twins of G with $\mathrm{thr}(u) \geq \mathrm{thr}(v)$. In addition, let S denote a target set with $v \in S$ and $u \notin S$. Then, $S' := (S \setminus \{v\}) \cup \{u\}$ is a target set for G.*

If the diameter of a given graph G is one, then G is a clique. By a simple exchange argument, we can see that there is a minimum-size target set for G that contains the vertices with the highest thresholds. In the following, we describe a linear-time algorithm that computes an optimal target set for a given clique.

By Observation 1, we can assume that the thresholds are upper-bounded by the cardinality n of the vertex set. Thus, we can sort the vertices by their thresholds in linear time. Once sorted, determining a minimum-size target set can be done in one pass: In the algorithm, we maintain an integer k representing the current target set size. Considering vertices v_i from lowest threshold to highest, we set $k := \max\{k, \min\{\text{thr}(v_i) - i, n - i\}\}$ until $i = n - k$. The remaining k vertices then form a minimum target set. In summary, we obtain the following.

Observation 8 (TSS on Cliques) *If G is a clique, then a minimum size target set for the instance (G, thr) can be computed in $O(n)$ time.*

3 Parameterization by the Diameter

Since many real-world (social) networks have small diameter (Girvan and Newman, 2002; Newman, 2003, 2010; Newman and Park, 2003), it is natural to investigate the influence of the parameter diameter on the complexity of TSS. This section is organized as follows.

First, we consider graphs with arbitrary threshold functions and bounded diameter. If the diameter of the input graph G is one (that is, G is a clique), then, by Observation 8, Target Set Selection can be solved in linear time. For graphs with diameter two, the close relationship to Dominating Set suggests that TSS is W[2]-hard (Downey et al, 2008). We confirm this intuition by proving W[2]-hardness of TSS with respect to the parameter "target set size" k even for split graphs of diameter two.

Second, we consider graphs with bounded diameter and restricted threshold functions. We show that Target Set Selection parameterized by the target set size k remains W[2]-hard when restricted to bipartite graphs with diameter four and constant or majority thresholds.

All our hardness proofs use parameterized reductions from the $W[2]$-hard Hitting Set problem and can be seen as refinements of the following basic reduction.

3.1 Basic Reduction

Our hardness proofs use parameterized reductions from Hitting Set (HS), which is defined as follows. Given a set family $\mathcal{F} = \{F_1, \ldots, F_m\}$ over a universe $U = \{u_1, \ldots, u_n\}$ and an integer $k \geq 0$, decide whether there is a *hitting set* $U' \subseteq U$ (that is, for every $1 \leq i \leq m$, it holds that $F_i \cap U' \neq \emptyset$) of size at most k. Hitting Set is NP-hard (Garey and Johnson, 1979) and W[2]-hard with respect to the parameter k (Downey and Fellows, 1999).

The following construction is the starting point of all our hardness reductions. Given a Hitting Set-instance consisting of a set family $\mathcal{F}$ over a universe U and a positive integer k, the bipartite *incidence graph* $G(U, \mathcal{F}) = (V_U \cup W_\mathcal{F}, E)$ is defined as follows. The first partition V_U contains a vertex for every element $u \in U$, that is, $V_U := \{v_u : u \in U\}$. Analogously, the second partition $W_\mathcal{F}$ contains a vertex for every subset, that is, $W_\mathcal{F} := \{w_F : F \in \mathcal{F}\}$. 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 $u \in F$ (see Figure 1

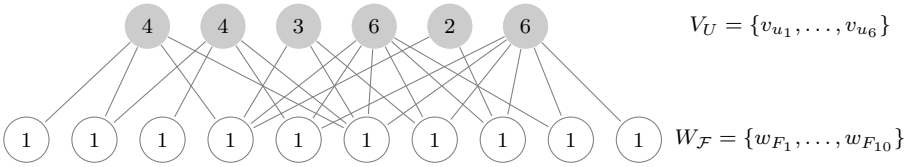

Fig. 1 Incidence graph for a Hitting Set-instance: $U = \{u_1, \ldots, u_6\}$ and $\mathcal{F} = \{F_1, \ldots, F_{10}\}$ with $F_1 = \{u_1\}$, $F_2 = \{u_1, u_2\}$, $F_3 = \{u_2\}$, $F_4 = \{u_1, u_3, u_4, u_5\}$, $F_5 = \{u_2, u_4, u_6\}$, $F_6 = \{u_1, u_2, u_3, u_4, u_6\}$, $F_7 = \{u_3, u_4, u_6\}$, $F_8 = \{u_4, u_5, u_6\}$, $F_9 = \{u_4, u_6\}$, $F_{10} = \{u_6\}$. The numbers in the vertices denote their thresholds.

for an example with given thresholds). Clearly, $G(U, \mathcal{F})$ can be constructed in polynomial time.

Lemma 1 *Let $(U, \mathcal{F})$ denote a* Hitting Set*-instance and let $G := G(U, \mathcal{F})$ denote the corresponding incidence graph. Moreover, let $\mathrm{thr}(v) = \deg_G(v)$ for all vertices $v \in V_U$ and $\mathrm{thr}(w) = 1$ for all $w \in W_{\mathcal{F}}$. Then $(U, \mathcal{F})$ has a hitting set of size at most k if and only if G has a target set of size at most k.*

Proof The "$\Rightarrow$"-direction follows by the simple observation that if $U' \subseteq U$ is a hitting set, then $V' := \{v_u : u \in U\}$ is a target set for G; since $\mathrm{thr}(w) = 1$ for all $w \in W_{\mathcal{F}}$, all vertices in $W_{\mathcal{F}}$ are activated in the first round. Then, in the second round, all vertices in V_U are activated.

For the "$\Leftarrow$"-direction, let S denote a target set for G of size at most k. By Observation 2 we can assume that $S \subseteq V_U$. We prove that $U' := \{u : v_u \in S\}$ is a hitting set by showing that every vertex in $W_{\mathcal{F}}$ is activated in the first round, and, hence, has a neighbor in S. Assume towards a contradiction that there is a vertex $w \in W_{\mathcal{F}}$ that is not activated in the first round, that is, $w \notin \mathcal{A}^1_{G,\mathrm{thr}}(S)$. Then, since $\mathrm{thr}(w) = 1$, no neighbor of w is contained in S. Note that to activate a vertex w at least one neighbor of w must be active before w. However, the activation of every neighbor v of w requires that w is active before v (since $\mathrm{thr}(v) = \deg(v)$). Both cases are mutually exclusive; a contradiction. □

3.2 Arbitrary Thresholds

If the thresholds are not restricted, then by Observation 8 we have a linear-time algorithm for graphs with diameter one. On the other hand, we can show $W[2]$-hardness with respect to the target set size k for graphs of diameter at least two, thus giving a complete complexity dichotomy for the parameter diameter.

In the following, we consider graphs with diameter at least two. Apparently, TSS becomes much harder in this case. Specifically, even restricted to diameter-two split graphs, TSS is NP-hard and $W[2]$-hard with respect to the parameter k.

Theorem 1 Target Set Selection *is NP-hard and $W[2]$-hard with respect to the parameter "target set size" k, even on split graphs with diameter two.*

Proof We present a parameterized reduction from Hitting Set, which is $W[2]$-complete (Downey and Fellows, 1999) with respect to the parameter "target set size". Given an HS-instance $(\mathcal{F}, U, k)$ consisting of a set family $\mathcal{F} = \{F_1, \ldots, F_m\}$

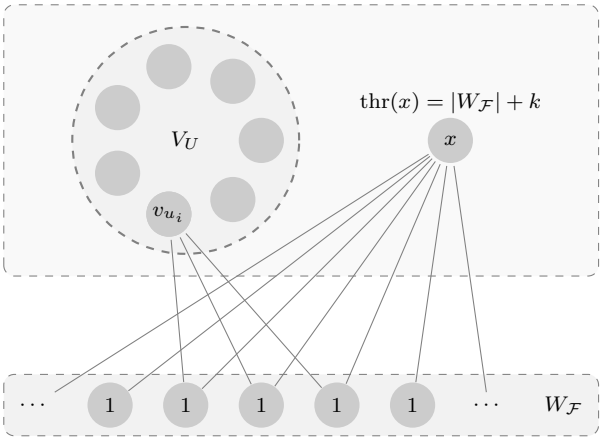

Fig. 2 A schematic picture of the constructed split graph. The vertices in the upper box form a clique, the vertices in the lower box form an independent set. The only way to activate all vertices with a target set of size k is to choose k vertices in V_U such that these k vertices activate all vertices in $W_{\mathcal{F}}$.

over a universe $U = \{u_1, \ldots, u_n\}$ and an integer $k \geq 0$, we construct a Target Set Selection-instance (G, thr, k) consisting of a graph $G = (V, F)$, a threshold function $\mathrm{thr} : V \to \mathbb{N} \cup \{0\}$ and k. To this end, recall the definition of the incidence graph $G(U, \mathcal{F}) = (V_U \cup W_{\mathcal{F}}, E)$ (see Section 2). The graph $G = (V, F)$ is constructed as follows. First, set $V := V_U \cup W_{\mathcal{F}}$ and $F := E$. Second, add a new vertex $x \notin (V_U \cup W_{\mathcal{F}})$ to G and connect x to all vertices in $W_{\mathcal{F}}$. Finally, make $V_U \cup \{x\}$ a clique. The thresholds are set as follows. For every subset-vertex $w_F \in W_{\mathcal{F}}$, set $\mathrm{thr}(w_F) := 1$ and for every element-vertex $v_u \in V_U$, set $\mathrm{thr}(v_u) := \deg_{G(U,\mathcal{F})}(v_u) + k + 1$. Finally, complete the construction by setting $\mathrm{thr}(x) := |W_{\mathcal{F}}| + k$. Since $W_{\mathcal{F}}$ is an independent set and $V_U \cup \{x\}$ is a clique, G is a diameter-two split graph, see Figure 2.

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 $S := \{v_u : u \in U'\}$ is a size-k target set for G. Since U' is a hitting set, all vertices in $W_{\mathcal{F}}$ are activated by S in the first round. Hence, x is activated in the second round. Finally, in the third round all vertices in $V_U \setminus S$ are activated since for every vertex in $V_U \setminus S$ all neighbors in $W_{\mathcal{F}}$ have been activated in the first round and x in the second.

"$\Leftarrow$": If (G, thr, k) is a yes-instance of TSS, then, by Observation 2, there is a target set S of size at most k that does not contain any vertex with threshold one. Hence, $S \subseteq V_U \cup \{x\}$. Clearly, we can assume without loss of generality that S contains exactly k vertices of $V_U \cup \{x\}$. Next, we show that $x \notin S$, and, hence, $S \subseteq V_U$. Assume towards a contradiction that $x \in S$. Then, $|S \cap V_U| = k - 1$. Let i denote the first round in which a vertex in $V_U \setminus S$ is activated, that is, $i := \min\{j : \mathcal{A}^j_{G,\mathrm{thr}}(S) \cap (V_U \setminus S) \neq \emptyset\}$. Moreover, let $v \in \mathcal{A}^i_{G,\mathrm{thr}}(S) \cap (V_U \setminus S)$. Note that, by

definition of i, it holds that $|\mathcal{A}^{i-1}_{G,\mathrm{thr}}(S) \cap (V_U \cup \{x\})| = k$. Hence:

$$\begin{aligned}|N_G(v) \cap \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)| =& |N_G(v) \cap \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S) \cap W_{\mathcal{F}}| \\ &+ |N_G(v) \cap \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S) \cap (V_U \cup \{x\})| \\ \leq& \deg_{G(U,\mathcal{F})}(v_u) + k \\ <& \deg_{G(U,\mathcal{F})}(v_u) + k + 1\end{aligned}$$

a contradiction. Hence, $S \subseteq V_U$. Finally, we show that $U' := \{u : v_u \in S\}$ is a hitting set for $\mathcal{F}$. To this end, we show that $W_{\mathcal{F}} \subseteq \mathcal{A}^1_{G,\mathrm{thr}}(S)$, and, hence, every subset-vertex v_F has a neighbor in S and, hence, is hit by U'. Assume that there exists a vertex $w_F \in W_{\mathcal{F}}$ with $w_F \notin \mathcal{A}^1_{G,\mathrm{thr}}(S)$. We show that $\mathcal{A}^2_{G,\mathrm{thr}}(S) = \mathcal{A}^1_{G,\mathrm{thr}}(S)$; since $w_F \notin \mathcal{A}^1_{G,\mathrm{thr}}(S)$, this is a contradiction to the fact that S is a target set for G. First, note that $(\mathcal{A}^1_{G,\mathrm{thr}} \cap (V_U \cup \{x\})) \setminus S = \emptyset$ since every vertex in $V_U \cup \{x\}$ has threshold at least $k+1$. Hence, $\mathcal{A}^1_{G,\mathrm{thr}}(S) \cap V_U = S$. Since $w_F \notin \mathcal{A}^1_{G,\mathrm{thr}}(S)$, it follows that $x \notin \mathcal{A}^2_{G,\mathrm{thr}}(S)$. Consider an arbitrary vertex $v_u \in V_U \setminus S$. Since $\mathcal{A}^1_{G,\mathrm{thr}}(S) \cap V_U = S$, it holds that $|\mathcal{A}^1_{G,\mathrm{thr}}(S) \cap N_G(v_u)| < \mathrm{thr}(v_u)$. Hence, $v_u \notin \mathcal{A}^2_{G,\mathrm{thr}}(S)$. Finally, note that $\mathcal{A}^2_{G,\mathrm{thr}}(S) \cap W_{\mathcal{F}} = \mathcal{A}^1_{G,\mathrm{thr}}(S) \cap W_{\mathcal{F}}$ since $W_{\mathcal{F}}$ is an independent set. In summary, $\mathcal{A}^2_{G,\mathrm{thr}}(S) = \mathcal{A}^1_{G,\mathrm{thr}}(S) \subsetneq V$; a contradiction to the fact that S is a target set for G. □

3.3 Restricted Thresholds

We show two computational hardness results for Target Set Selection on bipartite graphs of constant diameter and restricted threshold functions. We prove that TSS on bipartite graphs with diameter four remains NP-hard and $W[2]$-hard with respect to the parameter target set size k, even if all vertices have threshold two or all vertices respect the majority-threshold condition (see Section 2).

For both results, we modify the generic reduction given in Section 3.1 to match the respective restrictions.

Theorem 2 Target Set Selection *is NP-hard and $W[2]$-hard with respect to the parameter "target set size" k, even on bipartite graphs with diameter four and all vertices having threshold two.*

Proof We present a parameterized reduction from Hitting Set. 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 $(G, \mathrm{thr}_2, k+2)$ consisting of a graph $G = (V, F)$, the threshold function $\mathrm{thr}_2 : V \to \{2\}$, and the integer $k+2$. To this end, recall the definition of the incidence graph $G(U, \mathcal{F}) = (V_U \cup W_{\mathcal{F}}, E)$ (see Section 3.1). To construct G from $G(U, \mathcal{F})$, we keep the vertices in $W_{\mathcal{F}}$, replace the vertices in V_U with element-gadgets consisting of joined four-cycles and add some more gadgetry to activate these element-gadgets if all vertices corresponding to sets containing the corresponding element are active. Finally, we add a global shortcut-gadget that decreases the diameter. The details follow.

First, for each vertex $v_u \in V_U$, construct an element-gadget containing key vertices that, if active, activate all w_F with $u \in F$. Let $(w^0_u, w^1_u, \ldots, w^{s-1}_u)$ be an

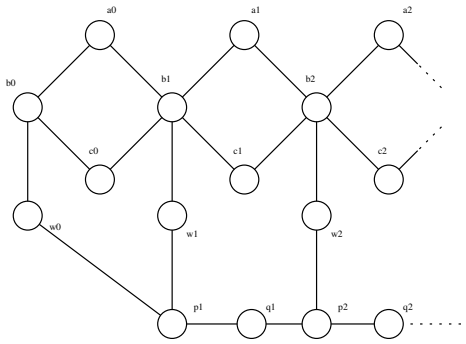

Fig. 3 A partial example of the element-gadget $\{a_u^i, b_u^i, c_u^i\}$ and the feedback-gadget $\{w_u^i, p_u^i, q_u^i\}$ for element u in the construction for Theorem 2.

arbitrary ordering of $N_{G(U,\mathcal{F})}(v_u)$. For each $0 \leq i < s$, the element-gadget consists of the path (a_u^i, b_u^i, c_u^i) and the additional edges $\{a_u^i, b_u^{i+1 \bmod s}\}$, $\{c_u^i, b_u^{i+1 \bmod s}\}$, and $\{b_u^i, w_u^i\}$. In this construction, let A denote the set of the vertices a_u^i and c_u^i with $v_u \in V_U$ and $0 \leq i < s$.

Second, for each vertex $v_u \in V_U$, construct a feedback-gadget that activates b_u^0 if all vertices w_F with $u \in F$ are active. The feedback-gadget consists of a path $(p_u^1, q_u^1, p_u^2, q_u^2, \ldots, p_u^{s-1}, q_u^{s-1})$, and the additional edges $\{q_u^{s-1}, b_u^0\}$, $\{p_u^1, w_u^0\}$, and $\{p_u^i, w_u^i\}$ for all $0 < i < s$. In this construction, let Q denote the set of all vertices q_u^i with $v_u \in V_U$ and $0 < i < s$. See Figure 3 for an illustration of the construction so far.

Finally, we construct a global shortcut-gadget with the purpose of decreasing the diameter of the constructed graph. The shortcut-gadget consists of the three new vertices x, x', and x'' with the edges $\{x, x'\}$ and $\{x, x''\}$. Furthermore, x is connected to all vertices in $A \cup W_{\mathcal{F}} \cup Q$. The thresholds of all vertices in the constructed graph G are set to two.

Note that $A \cup W_{\mathcal{F}} \cup Q \cup \{x', x''\}$ is a vertex cover and an independent set of G. Hence, G is bipartite and since x is connected to all vertices in this vertex cover, the diameter of G is at most four. Also note that $\deg(x') = \deg(x'') = 1$ and, thus, Observation 1 (Stubborn Vertices) implies that x' and x'' are contained in all target sets for G. The activation of x' and x'' also activates x, which, in turn, decreases the thresholds of all vertices adjacent to x by one. By Observation 5 (Chain Reduction), it suffices to show that the $\{x', x''\}$-reduced instance (G', thr', k) of $(G, \mathrm{thr}_2, k+2)$ is equivalent to $(\mathcal{F}, U, k)$.

In the following, we show that $(\mathcal{F}, U, k)$ is a yes-instance if and only if (G', thr', k) is a yes-instance. First, observe that for each $u \in U$, the set $\mathcal{A}_{G',\mathrm{thr}'}(\{b_u^0\})$ contains all vertices a_u^i, b_u^i, c_u^i, and w_u^i for all $0 \leq i < s$ and the set $\mathcal{A}_{G',\mathrm{thr}'}(\{w_u^0, \ldots, w_u^{s-1}\})$ contains all vertices p_u^i and q_u^i for all $0 < i < s$. Thus, we know that $b_u^0 \in \mathcal{A}_{G',\mathrm{thr}'}(\{w_u^0, \ldots, w_u^{s-1}\})$, since $q_u^{s-1}, w_u^0 \in \mathcal{A}_{G',\mathrm{thr}'}(\{w_u^0, \ldots, w_u^{s-1}\})$. Thus, both $\mathcal{A}_{G',\mathrm{thr}'}(\{b_u^0\})$ and $\mathcal{A}_{G',\mathrm{thr}'}(\{w_u^0, \ldots, w_u^{s-1}\})$ contain the entire element-gadget of v_u, all w_u^i with $0 \leq i < s$ and also the entire feedback-gadget of v_u.

"$\Rightarrow$": If $(\mathcal{F}, U, k)$ is a yes-instance of HS, then there is a size-k hitting set U' for $\mathcal{F}$. We show that $S := \{b_u^0 : u \in U'\}$ is a target set for G'. Note that for all vertices $x \in A \cup W_{\mathcal{F}} \cup Q$ we have $\mathrm{thr}'(x) = 1$.

Since U' is a hitting set, for each vertex $F \in \mathcal{F}$ there is some $u \in F$ with $b_u^0 \in S$. Then, by the above observations, $\mathcal{A}_{G',\mathrm{thr}'}(S)$ contains all vertices in $W_{\mathcal{F}}$ and, therefore, all vertices in feedback- and element-gadgets.
"$\Leftarrow$": If (G', thr', k) is a yes-instance of TSS, then, by Observation 2 (Lemming Vertices), there is a target set S of size at most k that does not contain any vertex with threshold one.

By the above observations and Observation 3, we can assume that if S contains a vertex of a feedback- or element-gadget of some v_u, then S contains b_u^0. Since $\mathrm{thr}'(w) = 1$ for all $w \in W_{\mathcal{F}}$, we can assume that $S \subseteq \{b_u^0 : u \in U\}$. Consider some $u \in U$ such that $b_u^0 \notin S$. Then, by construction, there is some j such that q_u^{s-1} is in $\mathcal{A}^j_{G',\mathrm{thr}'}(S)$ and b_u^0 is not.

We show by contradiction that $S' := \{u \in U : b_u^0 \in S\}$ is a solution for $(\mathcal{F}, U, k)$. If S' is not a hitting set, then there is some $F \in \mathcal{F}$ such that $b_u^0 \notin S$ for each $u \in F$. However, since $w_F \in \mathcal{A}_{G',\mathrm{thr}'}(S)$, there is some $u \in F$ and some j such that b_u^i with $w_u^i = w_F$ is in $\mathcal{A}^j_{G',\mathrm{thr}'}(S)$ but w_F is not. Since no vertex of the element-gadget of v_u is in S, there is also some $\ell \leq j$ such that q_u^{s-1} is in $\mathcal{A}^\ell_{G',\mathrm{thr}'}(S)$, but b_u^i is not. Then, by construction of the feedback-gadget, $p_u^i \in \mathcal{A}^\ell_{G',\mathrm{thr}'}(S)$ which requires $w_u^i \in \mathcal{A}^\ell_{G',\mathrm{thr}'}(S)$.

In summary, there are two integers j and ℓ such that b_u^0 is in $\mathcal{A}^j_{G',\mathrm{thr}'}(S)$ but w_F is not and w_F is in $\mathcal{A}^\ell_{G',\mathrm{thr}'}(S)$ but b_u^0 is not, a contradiction. □

Next, we show that TSS remains $W[2]$-hard for constant-diameter graphs even if we restrict the thresholds to respect the majority condition. To this end, we employ the following two lemmas stating that for each instance of TSS (on arbitrary graphs) we can find an equivalent instance with a diameter-four bipartite graph that respects the majority threshold condition. It uses two local modifications that allow us to modify the thresholds of any vertex. The first lemma shows that given a TSS-instance with arbitrary threshold function, one can build an equivalent instance with majority thresholds.

Lemma 2 *Let $I := (G, \mathrm{thr}, k)$ be an instance of* TSS*. Then there is an instance $I' = (G', \mathrm{thr}', k+1)$ of* TSS *such that* thr' *respects the majority threshold condition and I is a yes-instance if and only if I' is a yes-instance.*

Proof Let $V_>$ and $V_<$ be the set of vertices v of G with $\mathrm{thr}(v) > \lceil \deg_G(v)/2 \rceil$ and $\mathrm{thr}(v) < \lceil \deg_G(v)/2 \rceil$, respectively. The instance I' is constructed from I as follows. First, set $G' := G$. Then, to establish majority thresholds, proceed as follows.

For a vertex $v \in V_>$, add $2\,\mathrm{thr}(v) - \deg_G(v)$ degree-one vertices to the neighborhood of v. These vertices ensure that the "majority threshold" of v in G' equals its original threshold $\mathrm{thr}(v)$.

For each vertex $v \in V_<$, add $\deg_G(v) - 2\,\mathrm{thr}(v)$ vertices, denoted by V_v^*, to the neighborhood of v. Moreover, add another vertex x and make x adjacent to all vertices in $V^* := \bigcup_{v \in V_<} V_v^*$. Furthermore, add $|V^*| + 2(k+2)$ degree-one vertices $D := \{d_1, \ldots, d_{|V^*|+2(k+2)}\}$ to the neighborhood of x. This completes the construction of G'. Finally, set $\mathrm{thr}'(v) := \lceil \deg_{G'}(v)/2 \rceil$ for all $v \in V(G')$ to ensure majority thresholds. See Figure 4 for an example of the construction. Next, we argue that I and I' are equivalent.

First, observe that $\mathrm{thr}(v) = \mathrm{thr}'(v)$ for all vertices $v \in V_>$. Second, observe that x is contained in every target set for G' of size at most $k+1$; even if all

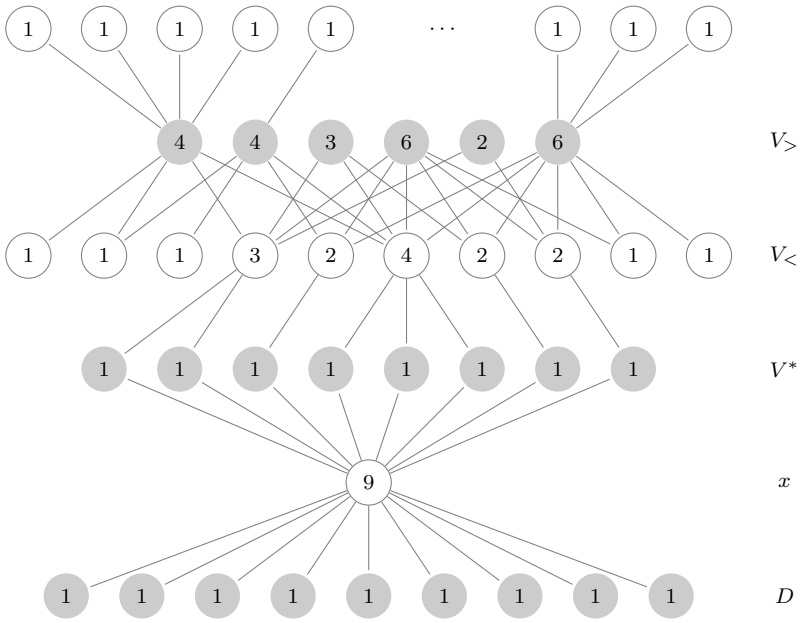

Fig. 4 An adjusted graph that respects majority threshold condition. The original graph is formed by the second and third row and is the same as in Figure 1.

vertices in $V_<$ are active (effectively decreasing the threshold of x to $(k+2)$) at least $(k+2)$ vertices from D must be contained in a target set in order to activate x (which is not possible for a target set of size at most $k+1$).

Let I'' denote the $\{x\}$-reduced instance of I'. By construction, in I'' the threshold of every vertex $v \in V$ equals its threshold in I (observe that activating x activates all vertices in V^*, which, in turn, implies that in I' the thresholds of the vertices in $V_<$ are decreased to their values in I). Moreover, since by Observation 2 we can assume that the newly introduced degree-one vertices adjacent to the vertices in $V_>$ are not contained in an optimal target set, it follows directly that I is a yes-instance if and only if I' is a yes-instance. □

Next, we show that one can transform any instance I of TSS into an equivalent instance with a bipartite graph of diameter four. The basic idea is as follows. First, we subdivide each edge as described in Observation 6 to obtain a bipartite graph. Second, by adding a new vertex x that is guaranteed to be in all optimal target sets, and connecting it to all vertices that were newly introduced in the subdivision process, we decrease the diameter to four while increasing k only by one and maintaining majority thresholds.

Lemma 3 *Let $I := (G, \mathrm{thr}, k)$ be an instance of* TSS. *Then, there is an instance $I' = (G', \mathrm{thr}', k+1)$ of* TSS *such that G' is bipartite, has diameter at most four, and I is a yes-instance if and only if I' is a yes-instance. Moreover, if I respects the majority threshold condition, then so does I'.*

Proof Given I, the new instance I' is built as follows. First, subdivide each edge as described in Observation 6. This ensures that the resulting graph is bipartite.

Let v_e denote the vertex that results by subdividing $e \in E$ and let $V_E := \{v_e : e \in E\}$. Furthermore, add a new vertex x and make x adjacent to all vertices in V_E. To ensure that x is contained in every target set of size at most $k+1$, add $|E|+2(k+2)$ degree-one vertices $D := \{d_1, \ldots, d_{|E|+2(k+2)}\}$ to the neighborhood of x and let G' be the resulting graph. Finally, set $\mathrm{thr}'(v_e) := \lceil \deg_{G'}(v_e)/2 \rceil = 2$ for all $v_e \in V_E$, $\mathrm{thr}'(x) := \lceil \deg_{G'}(x)/2 \rceil = |E|+k+2$, and $\mathrm{thr}'(d) := \lceil \deg_{G'}(d)/2 \rceil = 1$ for all $d \in D$. This completes the construction of I'.

First, observe that, if I respects the majority threshold condition, then, since the new thresholds all respect the majority threshold condition, I' respect the majority threshold condition as well. Moreover, note that the constructed graph is bipartite. One partition is given by $V_E \cup D$ and the other partition is $V \cup \{x\}$. Furthermore, the new vertex x is adjacent to all vertices in V_E. Thus the diameter is at most four.

Finally, we argue that I and I' are equivalent. To this end, first observe that x is contained in every target set for G' of size at most $k+1$; even if all vertices in V_E are active (effectively decreasing the threshold of x to $k+2$) at least $k+2$ vertices from D must be contained in a target set in order to activate x (which is not possible for a target set of size at most $k+1$). Second, observe that the graph of the $\{x\}$-reduced instance of I' is identical to the graph that results by subdividing every edge of G. Thus, Observation 6 implies that I is a yes-instance if and only if I' is a yes-instance. □

By combining Lemmas 1, 2 and 3, one directly obtains the following.

Theorem 3 Target Set Selection *is NP-hard and W[2]-hard with respect to parameter k, even on bipartite graphs with diameter four and majority thresholds.*

Note that the construction behind Theorem 3 is not limited to majority thresholds. It works for any threshold function that is linear in the degree. Hence, we have the following corollary.

Corollary 1 Target Set Selection *is NP-hard and W[2]-hard with respect to parameter k even on bipartite graphs with diameter four and any threshold function that is linear in the degree.*

4 Parameterization by the Cluster Editing Number

Target Set Selection is motivated by applications in social networks. Since social networks usually have clique-like substructures (communities), it is reasonable to consider a parameterization that measures the distance from the input graph to a graph consisting only of isolated cliques, a so-called "cluster graph" (Shamir et al, 2004). One such distance measure is the size ζ of a minimum-cardinality cluster editing set, that is, a smallest set of edges whose removal or addition results in a cluster graph. Another, similar distance measure is the size ξ of a minimum-cardinality cluster edge deletion set, that is, a smallest set of edges whose removal results in a cluster graph. By definition, $\zeta \leq \xi$.

For arbitrary thresholds, we show that TSS is fixed-parameter tractable with respect to the parameters ζ and ξ, by providing algorithms running in $O(16^{\zeta} \cdot m + n^3)$ and $O(4^{\xi} \cdot m + n^3)$ time, respectively. We present a linear-time computable

problem kernel whose number of vertices grows linearly with ζ when the thresholds are bounded by some constant. In fact, we can bound the number of vertices in this kernel by $2\zeta(t_{\max}+1)$, where $t_{\max}$ is the maximum threshold value occurring in the input.

Our elaborations highly depend on the notion of "critical cliques": a clique K in a graph is a *critical clique* if all its vertices have the same closed neighborhood and K is maximal with respect to this property. Computing all critical cliques of a graph can be done in linear time (McConnell and Spinrad, 1994). Note that critical cliques are always disjunct and, hence, the critical cliques of a graph form a partitioning of the vertices of the graph.

In this section, we assume that our input graph contains no isolated cliques: By Observation 8, we can find a minimum size target set in cliques in linear time. Hence, if our input graph contains isolated cliques, we can first solve these cliques optimal in linear time and then solve the remaining graph with the remaining value of k.

4.1 Arbitrary Thresholds

To present a strategy for solving TSS in case of arbitrary thresholds, we first introduce a data reduction rule that shrinks clique-like substructures. The key to showing fixed-parameter tractability is the observation that, in a graph reduced with respect to this data reduction rule, the search for an optimal target set can be restricted to a set of 4ζ vertices.

The description of the parameter-independent data reduction rule that shrinks clique-like substructures reads as follows. Consider a critical clique K in the input graph G and its neighbors $N_G(K)$. Note that $N_G(K)$ separates K from the rest of G. If activating all vertices in $N_G(K)$ is not enough to activate all vertices of K, then every target set of G has to contain some vertices of K. By Observation 7 (Twin Exchange), we can assume without loss of generality that those vertices have the highest thresholds among all vertices of K. These considerations lead to the following (see Figure 5 for an example).

Reduction Rule 2 *Let $I := (G, \mathrm{thr}, k)$ be an instance of* TSS *and let K be a critical clique in G. Moreover, let (G', thr', k') be the $N_G(K)$-reduced (see Definition 1) instance of $(G[N_G[K]], \mathrm{thr}, k)$ and let S denote an optimal target set for (G', thr'). Then, replace I with the S-reduced instance of I.*

Herein G' is a clique and, hence, an optimal target set S for (G', thr') can be computed in linear time (see Section 3). Note that this reduction rule is parameter-independent, that is, the reduction rule is executable without knowledge about the parameter value.

Lemma 4 *Reduction Rule 2 is correct and can be exhaustively applied in $O(n \cdot (n+m))$ time.*

Proof The correctness follows from the discussion before the presentation of Reduction Rule 2.

Next, we upper-bound the running time. All critical cliques can be found in linear time (McConnell and Spinrad, 1994). Let $K_1, \ldots, K_\ell$ denote the critical cliques.

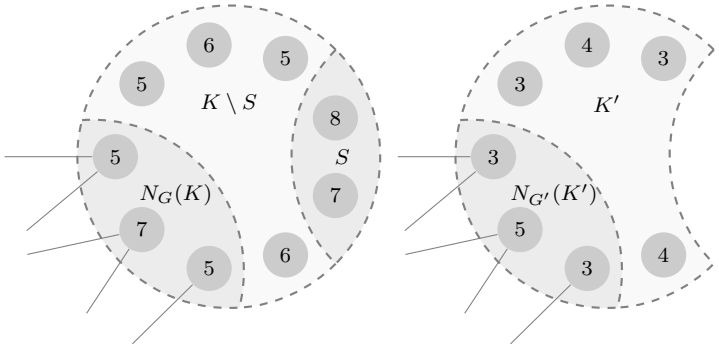

Fig. 5 Example for Reduction Rule 2. Left: A critical clique K of six vertices and its three neighbors. The two vertices with the highest threshold (vertices in S) have to be in the target set. Right side: The S-reduced instance (G', thr', k') of $(G[N_G[K]], \mathrm{thr}, k)$ with the remaining critical clique K' and the three neighbors.

Moreover, observe that all sets $N_G(K_1), \ldots, N_G(K_\ell)$ can be computed in $O(n+m)$ by one iteration over the edge set. The $N_G(K_i)$-reduced instance $(G_i', \mathrm{thr}', k')$ of $(G[N_G[K_i]], \mathrm{thr}, k)$ can be computed in linear time. Since K_i is a critical clique, G_i' is a clique. Hence, an optimal target set for (G_i', thr') is computable in $O(|K_i|)$ time. Thus, one application of Reduction Rule 2 needs linear time.

After one application of Reduction Rule 2 the critical cliques may change and thus have to be recomputed. However, each application of Reduction Rule 2 removes at least one vertex. Hence, Reduction Rule 2 can be exhaustively applied in $O(n \cdot (n+m))$ time. □

Having reduced the input instance with respect to Reduction Rule 2, we claim that we can limit our considerations to a small subset of vertices of the remaining graph. To see this, consider a cluster editing set E' of a graph G and let $V(E')$ denote the vertices incident to the edges in E' (we call the vertices in $V(E')$ *affected* by E'). Since $|E|' = \zeta$ we have $|V(E')| \leq 2\zeta$.

Let K be a critical clique of G and $V(K)$ the vertices of K. Let

$$\mathcal{K} := \{K \,:\, K \text{ is a critical clique in } G \text{ and } V(K) \cap V(E') = \emptyset\}$$

be the set of all critical cliques in G whose vertices are disjoint to $V(E')$. By the next lemma, it follows that $\mathcal{K}$ contains all vertices in $V \setminus V(E')$.

Lemma 5 *Let K be a critical clique. Then either $V(K) \subseteq V(E')$ or $V(K) \cap V(E') = \emptyset$.*

Proof Assume towards a contradiction that there is a critical clique K' with $V(K') \cap V(E') \neq \emptyset$ and $V(K') \setminus V(E') \neq \emptyset$. Let $K_1, \ldots, K_\ell$ denote the isolated cliques in the cluster graph $G' := (V, (E \setminus E') \cup (E' \setminus E))$. Let $v \in V(K') \setminus V(E')$. Since $v \notin V(E')$, it follows that $N_G[v] = V(K_i)$ for an i, $1 \leq i \leq \ell$. Let $u \in V(K') \cap V(E')$. Since $u \in V(E')$, it follows that $N_G[u] \neq V(K_j)$ for any j, $1 \leq j \leq \ell$. Note that $u \in V(K')$ and $v \in V(K')$ and, hence, $V(K_i) = N_G[v] = N_G[u] \neq V(K_i)$, a contradiction. □

We define K^{high} as the set of the $|N_G(K)|$ vertices of a critical clique K with the highest thresholds. If $|N_G(K)| > |V(K)|$, then set $K^{\text{high}} := V(K)$. Now, we define $V_{E'}^{\text{high}} := \bigcup_{K \in \mathcal{K}} K^{\text{high}}$. Next, we show that it suffices to limit our considerations to $V(E') \cup V_{E'}^{\text{high}}$.

Lemma 6 *Let $I := (G, \text{thr}, k)$ denote a yes-instance of* TSS *reduced with respect to Reduction Rule 2 and let E' denote an optimal cluster editing set of G. Then, there exists an optimal target set S for (G, thr) with $S \subseteq V(E') \cup V_{E'}^{\text{high}}$.*

Proof Let S be an optimal target set for (G, thr). Since (G, thr) is reduced with respect to Reduction Rule 2, $V(E')$ is a target set for (G, thr) and, hence, $|S| \leq |V(E')| \leq 2\zeta$. Thus, there is no critical clique $K \in \mathcal{K}$ with more than $|N_G(K)|$ vertices in S. Since all vertices in a critical clique have the same neighborhood, by Observation 7 we can assume without loss of generality that only the vertices with the highest thresholds are in S. Hence, $S \subseteq V(E') \cup V_{E'}^{\text{high}}$. □

The next lemma bounds the size of $V_{E'}^{\text{high}}$. Since $|V(E')| \leq 2\zeta$, the search space is bounded by the parameter (also referred to as "weak kernel" in recent literature (Jiang et al, 2010)) and, hence, fixed-parameter tractability follows.

Lemma 7 $|\mathcal{K}| \leq |V_{E'}^{\text{high}}| \leq 2\zeta$.

Proof Remember that G does not contain isolated cliques (see assumption in the beginning of this section). Thus, in each $K \in \mathcal{K}$ at least one vertex is in $V_{E'}^{\text{high}}$ and, hence, $|\mathcal{K}| \leq |V_{E'}^{\text{high}}|$. It remains to show that $|V_{E'}^{\text{high}}| \leq 2\zeta$.

To this end, let $K_1, \ldots, K_\ell$ denote the isolated cliques in the cluster graph $G' := (V, (E \setminus E') \cup (E' \setminus E))$. Moreover, let $U := V \setminus V(E')$ denote the unaffected vertices and let $X_i := V(K_i) \cap U$ and $Y_i := V(K_i) \cap V(E')$ for all $1 \leq i \leq \ell$. Observe that the X_i's form the vertices of the critical cliques in $\mathcal{K}$ implying $\sum_{K \in \mathcal{K}} |N_G(K)| = \sum_{i=1}^{\ell} |Y_i|$. Hence:

$$|V_{E'}^{\text{high}}| = \sum_{K \in \mathcal{K}} |K^{\text{high}}| = \sum_{K \in \mathcal{K}} |N_G(K)| = \sum_{i=1}^{\ell} |Y_i| = |V(E')| \leq 2\zeta$$

□

By Lemma 6, it holds that there exists an optimal target set S with $S \subseteq V(E') \cup V_{E'}^{\text{high}}$. Remember that $|V(E')| \leq 2\zeta$ and, by Lemma 7, $|V_{E'}^{\text{high}}| \leq 2\zeta$. Hence, an optimal target set $S \subseteq V(E') \cup V_{E'}^{\text{high}}$ can be found by systematically checking every subset of at most $|V(E') \cup V_{E'}^{\text{high}}| \leq 4\zeta$ vertices. This leads to the following theorem.

Theorem 4 Target Set Selection *can be solved in $O(16^{\zeta} \cdot m + n^3)$ time, where ζ denotes the cluster editing number of the input graph.*

Proof We assume that the input instance is reduced with respect to Reduction Rule 2. To solve TSS, proceed as follows. In a first step, compute an optimal cluster editing set E'. Then, check each subset of $V(E') \cup V_{E'}^{\text{high}}$ for being a target set for (G, thr, k) and choose one of minimum cardinality.

The correctness follows directly by Lemma 6: since (G, thr, k) is reduced with respect to Reduction Rule 2 there is an optimal target set $S \subseteq V(E') \cup V_{E'}^{\mathrm{high}}$. For the running time note the following. By Lemma 4, a reduced instance can be computed in $O(n \cdot m)$ time. Moreover, note that an optimal cluster editing set can be found in $O(1.62^{\zeta} + \zeta^2 + n + m)$ time (Böcker, 2011). Finally, by Lemma 6 and Lemma 7 we have $|V(E') \cup V_{E'}^{\mathrm{high}}| \leq 4\zeta$. Hence, given E', one can find an optimal target set by checking every $S \subseteq V(E') \cup V_{E'}^{\mathrm{high}}$ in $O(2^{4\zeta} \cdot m + n^3)$ time. □

Cluster Edge Deletion Number. For Target Set Selection parametrized by the weaker[8] parameter "cluster edge deletion number ξ" we show that there is an optimal target set consisting only of affected vertices. Here the affected vertices $V(E')$ are vertices incident to edges from the cluster edge deletion set E'. Since $|V(E')| \leq 2\xi$, this directly implies fixed-parameter tractability with respect to ξ with a "better" running time bound compared to the stronger parameter cluster editing number. The details follow.

Lemma 8 *Let $I := (G, \mathrm{thr}, k)$ denote a yes-instance of* TSS *reduced with respect to Reduction Rule 2 and let E' denote an optimal cluster edge deletion set of G. Then, there exists an optimal target set S for I with $S \subseteq V(E')$.*

Proof Let S be an optimal target set for (G, thr) with $S \setminus V(E') \neq \emptyset$. Moreover, let $U := V \setminus V(E')$ denote the unaffected vertices. We show that there is also an optimal target set for G that contains only vertices of $V(E')$.

Let $x \in U \cap S$ be arbitrarily chosen and let K denote the isolated clique in $(V, E \setminus E')$ with $x \in K$. Let $X = K \cap U$ and $Y = K \setminus U$. Clearly, K is a clique in G and $Y = N_G(X)$. Moreover, observe that X is a critical clique in G with $x \in X$. We show that there is a vertex $y \in Y$ such that $(S \setminus \{x\}) \cup \{y\}$ is a target set for G. By an iterative application of this exchange argument, we can conclude that there is an optimal solution containing only affected vertices.

Let (G', thr', k') denote the $S' := S \setminus \{x\}$−reduced instance of (G, thr, k). Since S is an optimal target set and the instance is reduced with respect to Reduction Rule 2, we can assume that $x \in V(G')$ and $Y \cap V(G') \neq \emptyset$. Clearly, $\{x\}$ is a target set for (G', thr') and, by Observation 5, for every target set S'' for (G', thr') the set $(S \setminus \{x\}) \cup S''$ is a target set for (G, thr).

Let $X' := V(G') \cap X$ and $Y' := V(G') \cap Y$. By the above discussion, we have that $X' \neq \emptyset$ and $Y' \neq \emptyset$. Moreover, since $X' \cup Y'$ is a clique and Y' separates X' from the rest of G', we know that $N_{G'}[x] \subseteq N_{G'}[y]$ for all $y \in Y'$.

Let ℓ denote the smallest integer such that $Y' \subseteq \mathcal{A}^{\ell}_{G', \mathrm{thr}'}(\{x\})$. Let $y \in Y' \setminus \mathcal{A}^{\ell-1}_{G', \mathrm{thr}'}(\{x\})$ (that is, y is activated by $\{x\}$ in the ℓ^{th} round). We show that $\{y\}$ is a target set for (G', thr') and, hence, $(S \setminus \{x\}) \cup \{y\}$ is a target set for (G, thr). To this end, let $B(x, i) := \mathcal{A}^{i}_{G', \mathrm{thr}'}(\{x\}) \setminus \{x\}$ and $B(y, i) := \mathcal{A}^{i}_{G', \mathrm{thr}'}(\{y\}) \setminus \{y\}$. More specifically, we show by induction on the number of activation rounds that for all $0 \leq i < \ell$ it holds that $B(x, i) \subseteq B(y, i)$. Then, from $B(x, \ell - 1) \subseteq B(y, \ell - 1)$, we can conclude that $Y' \subseteq \mathcal{A}^{\ell}_{G', \mathrm{thr}'}(\{y\})$. Since G' is reduced with respect to Reduction Rule 2, Y' is a target set for $G'[X' \cup Y']$. Thus, we have that $x \in \mathcal{A}^{r}_{G', \mathrm{thr}'}(\{y\})$ for some $r \geq \ell$ and, hence, $\{y\}$ is a target set for G'.

[8] The cluster edge deletion number ξ is lower-bounded by the cluster editing number ζ, that is, $\xi \geq \zeta$.

For the base case, observe that $B(x,0) = B(y,0) = \emptyset$. Next, we provide the induction step, that is, we show that $B(x,i-1) \subseteq B(y,i-1)$ implies $B(x,i) \subseteq B(y,i)$ for all $0 \leq i < \ell$. Assume towards a contradiction that $B(x,i-1) \subseteq B(y,i-1)$ but $B(x,i) \setminus B(y,i) \neq \emptyset$ and let $v \in B(x,i) \setminus B(y,i)$. Since by definition of ℓ we know that $y \notin \mathcal{A}^{i}_{G',\mathrm{thr}'}(\{x\})$ for all $i < \ell$ and $N_{G'}[x] \subseteq N_{G'}[y]$, the vertex v must have more neighbors in $B(x,i-1)$ than in $B(y,i-1)$ in order to be activated by $B(x,i-1) \cup \{x\}$ but not by $B(y,i-1) \cup \{y\}$. This is a contradiction to $B(x,i-1) \subseteq B(y,i-1)$.

Thus, $B(x,\ell-1) \subseteq B(y,\ell-1)$ and, by the same argument as in the proof of the induction step, it follows that $Y' \subseteq \mathcal{A}^{\ell}_{G',\mathrm{thr}'}(\{y\})$. Finally, x is activated because Y' activates all vertices in X'. All in all, $\{y\}$ is a target set for G' and $(S \setminus \{x\}) \cup \{y\}$ is a target set for G. □

By Lemma 8, an optimal target set can be found by systematically checking every subset of the at most 2ξ vertices affected by an optimal cluster edge deletion set, leading directly to the following theorem.

Theorem 5 Target Set Selection *can be solved in* $O(4^{\xi} \cdot m + n^3)$ *time, where* ξ *denotes the cluster edge deletion number of the input graph.*

Proof First, we describe the solving strategy. We assume that the input instance is reduced with respect to Reduction Rule 2. To solve TSS, proceed as follows. In a first step, compute an optimal cluster edge deletion set E'. Then, check each subset of $V(E')$ for being a target set for (G, thr).

The correctness follows directly by Lemma 8; since (G, thr) is reduced with respect to Reduction Rule 2 there is an optimal target set $S \subseteq V(E')$. For the running time note the following. By Lemma 4, a reduced instance can be computed in $O(n \cdot m)$ time. Moreover, note that an optimal cluster edge deletion set can be found in $O(1.42^{\xi} + nm)$ time (Böcker and Damaschke, 2011). Finally, note that $|V(E')| \leq 2\xi$. Hence, given E', one can find an optimal target set by checking every $S \subseteq V(E')$ in $O(2^{2\xi} \cdot m + n^3)$ time. □

Note that all FPT-results in this section rely on checking all subsets of a size-bounded vertex set. Finding non-brute-force algorithms for parameter cluster editing number or cluster edge deletion number remains as a task for future work.

4.2 Restricted Thresholds

In the previous section, Reduction Rule 2 helped us shrink the set of vertices that have to be considered for an optimal target set. While this constitutes a "weak kernel" and helps solve Target Set Selection, we could not derive a polynomial-size problem kernel in the classical sense. As it turns out, the obstacle is not being able to bound the thresholds of vertices. However, if the instances that need to be solved already exhibit an upper bound on the maximum threshold, then we can compute a polynomial size problem kernel. In the following, we present a data reduction rule shrinking large critical cliques until their size can be bounded by this maximum threshold $t_{\max}$. Thereby, we obtain a problem kernel for TSS with $2\zeta(t_{\max}+1)$ vertices, which is linear in ζ if the thresholds of the input are bounded by a constant.

Consider a critical clique K containing at least $t_{\max}+1$ vertices. Informally speaking, it is sufficient to keep the $t_{\max}$ vertices of K with the smallest thresholds, since if these are active, then all vertices in $N_G[K]$ are activated.

Reduction Rule 3 *Let (G, thr, k) be an instance of* TSS *and let $t_{\max}$ denote the maximum threshold of this instance. Furthermore, let K be a critical clique in G with $|K| > t_{\max}$ and let $K^{\mathrm{high}}_{t_{\max}}$ denote the $|K| - t_{\max}$ vertices inside K with highest thresholds. Then, delete the vertices in $K^{\mathrm{high}}_{t_{\max}}$ from G.*

Lemma 9 *Reduction Rule 3 is correct and can be exhaustively applied in linear time.*

Proof We first show the correctness of the reduction rule: $(G = (V, E), \mathrm{thr}, k)$ is a yes-instance if and only if the reduced instance $(G' = (V', E'), \mathrm{thr}', k)$ is a yes-instance.

"$\Leftarrow$": Let S be a target set for (G', thr', k). Let $v \in V \setminus V'$ be a vertex removed by the application of the data reduction rule. Then v has at least $t_{\max}$ neighbors in K that are not removed. Since S activates all the remaining $t_{\max}$ vertices in K and $\mathrm{thr}(v) \leq t_{\max}$, the vertex v also gets activated. Hence, S is also a target set for (G, thr, k).

"$\Rightarrow$": In the following, we show that if S contains a vertex $v \in K^{\mathrm{high}}_{t_{\max}}$, then we can exchange v for the vertex u with the highest threshold in $K \setminus (S \cup K^{\mathrm{high}}_{t_{\max}})$ in S. Since $u \notin S$, there must be some $j > 0$ such that u is activated in the j^{th} activation round. Since there is no vertex in $K \setminus (S \cup K^{\mathrm{high}}_{t_{\max}})$ with higher threshold than u, we know that all vertices of K that are not in $K^{\mathrm{high}}_{t_{\max}}$ are active in activation round j. Since v is active in each activation round i with $i < j$, and u and v have the same closed neighborhood, we do not change $\mathcal{A}^i_{G,\mathrm{thr}}(S) \setminus \{u, v\}$ by replacing v with u in S. Thus, all vertices in $K \setminus K^{\mathrm{high}}_{t_{\max}}$ are active in the j^{th} activation round. Since $|K \setminus K^{\mathrm{high}}_{t_{\max}}| = t_{\max}$, all vertices in $K^{\mathrm{high}}_{t_{\max}}$ are active in the activation round $j+1$. By iteratively applying this argument, we arrive at a target set S with $S \cap K^{\mathrm{high}}_{t_{\max}} = \emptyset$. Hence, S is a target set for the reduced instance (G', thr', k).

For the running time, note that all critical cliques can be found in $O(n+m)$ time (McConnell and Spinrad, 1994). Let $K_1, \ldots, K_\ell$ denote the critical cliques of G. We can sort the vertices in these critical cliques by their thresholds in $O(|K_i|)$ time using bucket sort in order to determine the vertices that can be deleted. Hence, for all critical cliques we need $O(\sum_{i=1}^{\ell} |K_i|) = O(n)$ time. Clearly, deleting these vertices is doable in $O(n+m)$ time. Note that deleting some (but not all) vertices of a critical clique does not change other critical cliques. Hence, the critical cliques have to be computed only once. Thus, the total running time for the exhaustive application of Reduction Rule 3 is linear. □

By Lemma 7, the number of critical cliques that are disjoint to $V(E')$ is upper-bounded by 2ζ and each critical clique contains at most $t_{\max}$ vertices. Thus, the number of vertices in a reduced graph is at most $2\zeta(t_{\max}+1)$.

Theorem 6 Target Set Selection *admits a problem kernel with $2\zeta(t_{\max}+1)$ vertices, where ζ and $t_{\max}$ denote the cluster editing number and the maximum threshold value of the input graph, respectively. The kernelization runs in linear time.*

Proof Let $(G = (V, E), \mathrm{thr})$ denote an instance that is reduced with respect to Reduction Rule 3. By Lemma 7 there are at most 2ζ critical cliques disjoint to $V(E')$.

Hence, $|V| = |V(E')| + \sum_{K \in \mathcal{K}} |K| \leq 2\zeta + 2\zeta t_{\max} = 2\zeta(t_{\max} + 1)$. The running time follows directly from Lemma 9. □

5 Parameterization by the Vertex Cover Number

The vertex cover number τ of a graph G denotes the cardinality of an optimal vertex cover of G (that is, a minimum-size vertex set such that every edge of G has at least one endpoint in this set). Since deleting all vertices in a vertex cover results in a graph without edges, a vertex cover of a graph G is a feedback vertex set for G as well. Moreover, it is a well-known fact that the feedback vertex set number is an upper bound on the treewidth. Hence, the vertex cover number is an upper bound on the treewidth, as well. Ben-Zwi et al (2011) have shown that TSS is W[1]-hard for the combined parameter "treewidth and target set size". Indeed, their reduction shows that TSS is W[1]-hard even for the combined parameter "feedback vertex set number and target set size". These hardness results motivate the study of TSS parameterized by parameters larger than the feedback vertex set number as, for example, the vertex cover number (Bodlaender et al, 2011a; Fellows et al, 2008; Fiala et al, 2011).

5.1 Arbitrary Thresholds

If the input graph G is reduced with respect to Reduction Rule 1 from Section 2, then the threshold of each vertex is bounded by its degree. Thus, each vertex can be activated by activating all its neighbors. This implies that a vertex cover of G is also a target set for G and, hence, the target set size k is at most the vertex cover number τ of G. In this sense, the vertex cover number τ is a "weaker" parameter than the target set size k. This allows us to develop fixed-parameter algorithms with respect to the parameter τ.

Let $G = (V, E)$ denote a graph and let Z denote a vertex cover of G. Then, $I := V \setminus Z$ is an independent set. The key for showing fixed-parameter tractability of TSS with respect to the vertex cover number τ is to upper-bound the number of vertices in I that are candidates for an optimal target set. To this end, vertices in I with an identical neighborhood are of particular interest.

The vertices of I are contained in at most $2^{|Z|}$ critical independent sets (see Definition 2) – one for each subset of Z. Observation 7 (Twin Exchange) from Section 2 allows us to focus on a restricted number of vertices for each critical independent set when looking for an optimal target set.

In the following, we assume that the vertices of the input graph are sorted decreasingly by their thresholds. By Observation 7, we can directly conclude that there is an optimal target set that contains for each critical independent set I only vertices from the set of k vertices with highest threshold. Moreover, we can bound the number of critical independent sets by a function of τ, leading to fixed-parameter tractability of TSS with respect to τ.

Theorem 7 Target Set Selection *can be solved in* $O(2^{(2^\tau+1)\cdot\tau} \cdot m)$ *time, where* τ *denotes the vertex cover number of the input graph.*

Proof Let $(G = (V, E), \mathrm{thr}, k)$ denote the input instance of TSS. In the following, we assume that G is reduced with respect to Reduction Rule 1 from Section 2.

The algorithm works as follows. In a first phase, it computes an optimal vertex cover Z of G (this can be done in $O(1.2738^\tau + \tau n)$ time (Chen et al, 2010)). If $k \geq |Z|$, then the algorithm returns Z. Otherwise, in a second phase the algorithm computes a set C of at most $(2^\tau + 1) \cdot \tau$ vertices for which there exists an optimal target set S with $S \subseteq C$. The set C is computed as follows: For each critical independent set, C contains k vertices with highest threshold (for a critical independent set of cardinality at most k all its vertices are in C). In the third phase, the algorithm systematically checks every size-k subset of C for being a target set. If no target set is found, then it rejects the instance.

Next, we show that the presented algorithm finds a target set of size at most k for G, if it exists. For the correctness of the first phase note that, since G is reduced with respect to Reduction Rule 1, any vertex cover of G is also a target set for G. For the correctness of the other phases note that, by iteratively applying Observation 7, we can conclude that there exists an optimal target set S with $S \subseteq C$. Thus, the correctness of the algorithm follows by the fact that it checks every size-k subset of C.

Finally, we give an upper bound on the running time of the algorithm. The running time of the first phase is dominated by the computation of an optimal vertex cover which is doable in $O(1.2738^\tau + \tau n)$ time (Chen et al, 2010). Moreover, the second phase of the algorithm needs $O(n + m)$ time, which can be seen as follows. All critical independent sets of G can be computed in $O(n + m)$ time (McConnell and Spinrad, 1994). Let $I_1, \ldots, I_\ell$ denote the critical independent sets of G. Then, sorting all critical independent sets by their thresholds with bucket sort, one can compute C in $O(\sum_{i=1}^{\ell} |I_i|) = O(n + m)$ time.

For the running time bound of the third phase we first show that $|C| \leq 2^\tau \cdot \tau + \tau$. Note that the vertices in $V \setminus Z$ are contained in at most 2^τ critical independent sets (one for every subset of Z). Since C contains at most $k \leq \tau$ vertices from each of these critical independent sets, C contains at most $2^\tau \cdot \tau$ vertices from $V \setminus Z$. In addition, C contains all vertices from Z, summing up to at most $2^\tau \cdot \tau + \tau$ in total. This observation allows us to bound the running time of the third phase by $O(2^{|C|} \cdot (n + m)) = O(2^{(2^\tau+1)\cdot\tau} \cdot (n + m))$ since the test whether a given set of vertices is a target set is doable in $O(n + m)$ time. Altogether, the running time of the algorithm is bounded by $O(2^{(2^\tau+1)\cdot\tau} \cdot (n + m))$. □

5.2 Restricted Thresholds

Next, we show that TSS admits a problem kernel with $O(2^\tau \cdot t_{\max})$ vertices. Moreover, we show that TSS parameterized by the vertex cover number presumably does not admit a polynomial-size problem kernel even for majority thresholds.

First, we present a problem kernel for the combined parameter "vertex cover number τ and maximum threshold $t_{\max}$". To bound the number of vertices in a reduced instance, we need (besides Reduction Rule 1 from Section 2) one further data reduction rule that shrinks large critical independent sets. For every critical independent set of size at least $t_{\max}$, this rule removes all but $t_{\max}$ vertices with smallest thresholds.

Reduction Rule 4 *Let (G, thr, k) denote a* TSS*-instance reduced with respect to Reduction Rule 1 and let I denote a critical independent set of G with $|I| > t_{\max}$. Then delete $|I| - t_{\max}$ highest-threshold vertices of I.*

Lemma 10 *Reduction Rule 4 is correct. Moreover, one application of Reduction Rule 4 takes $O(n+m)$ time.*

Proof Let $G' = (V', E')$ denote the graph that results from applying Reduction Rule 4 to $(G = (V, E), \mathrm{thr}, k)$. Moreover, let H denote $|I| - (t_{\max})$ vertices of I with highest thresholds and let $I' := I \setminus H$. Note that $G' = G[V \setminus H]$. For the correctness we show that (G, thr, k) is a yes-instance if and only if (G', thr, k) is a yes-instance.
"$\Leftarrow$": Let S' be a target set for G' of size at most k. We show that S' is a target set for G as well. Since $G' = G[V \setminus H]$, all vertices in $V \setminus H$ are activated by S' in G. Moreover, since $\mathrm{thr}(v) \leq \deg_G(v)$ for all vertices $v \in V$ (recall that G is reduced with respect to Reduction Rule 1), the vertices in I are activated by S'.
"$\Rightarrow$": Let S denote an optimal target set of size at most k for G. We show how to construct from S an optimal target set S' for G'. Let $\ell := |S \cap I|$. By Observation 7 from Section 2 we can assume that S contains the ℓ vertices with largest thresholds from I.

First, observe that every optimal target set for G contains at most $t_{\max}$ vertices of I: Assume towards a contradiction that S contains at least $t_{\max} + 1$ vertices from I. Let $v \in S \cap I$ be arbitrarily chosen. Then $S^* := S \setminus \{v\}$ is a solution for G since $|S^* \cap I| \geq t_{\max}$ and hence $N(I) \subseteq \mathcal{A}^1_{G,\mathrm{thr}}(S^*)$, and, as a consequence, $\mathcal{A}^2_{G,\mathrm{thr}}(S^*) = \mathcal{A}^2_{G,\mathrm{thr}}(S)$.

By this observation, $\ell \leq t_{\max}$. Let S'' denote the ℓ vertices with highest thresholds from I'. We show that $S' := (S \setminus I) \cup S''$ is a solution for G'. For ease of presentation, we introduce the following notation. Let $B(i) := \mathcal{A}^i_{G,\mathrm{thr}}(S) \setminus I$ and $B'(i) := \mathcal{A}^i_{G',\mathrm{thr}'}(S') \setminus I'$. Moreover, let $I(i) := I \cap \mathcal{A}^i_{G,\mathrm{thr}}(S)$ and $I'(i) := I' \cap \mathcal{A}^i_{G',\mathrm{thr}'}(S')$. We prove by induction that $B(i) \subseteq B'(i)$ for all $i \in \mathbb{N}$. This shows that S' is a target set for G' since then all vertices in $V' \setminus I'$ and, hence, all vertices in I' are activated since $\mathrm{thr}(v) \leq \deg_{G'}(v)$ for all $v \in I'$.

For the induction base, we show that $B(0) = B'(0)$ and $B(1) = B'(1)$. Obviously, $B(0) = B'(0)$ since $(V \setminus I) \cap S = (V' \setminus I') \cap S'$. Moreover, $B(1) = B'(1)$, since every vertex in $V' \setminus N_{G'}[I'] = V \setminus N_G[I]$ has identical neighborhoods in G and G', and, by the construction of S', every vertex in $N_G(I)$ has the same number of neighbors in $I \cap \mathcal{A}^0_{G,\mathrm{thr}}(S)$ and $I' \cap \mathcal{A}^0_{G',\mathrm{thr}'}(S')$.

Next, we show the induction step, that is, we prove that from $B(i-2) \subseteq B'(i-2)$ and $B(i-1) \subseteq B'(i-1)$ it follows that $B(i) \subseteq B'(i)$ for all $i \geq 2$. Let $v \in B(i)$ be arbitrarily chosen. We show that $v \in B'(i)$. If $v \in V' \setminus N[I']$, then this follows directly by the observation that, in this case, v has an identical neighborhood in G and G', and, by the induction hypothesis, at least as many neighbors in $\mathcal{A}^{i-1}_{G',\mathrm{thr}'}(S')$ as in $\mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)$. Hence, in the following we consider the case that $v \in N_G(I)$. First, note that if $|I'(i-1)| \geq t_{\max}$, then $v \in B'(i)$. That is, it remains to prove that $v \in B'(i)$ if $|I'(i-1)| < t_{\max}$. To this end, we show that if $|I'(i-1)| < t_{\max}$, then $|I'(i-1)| \geq |I(i-1)|$ and, hence, v has at least as many neighbors in $\mathcal{A}^{i-1}_{G',\mathrm{thr}'}(S')$ as in $\mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)$. By the induction hypothesis it holds that $B(i-2) \subseteq B'(i-2)$. Moreover, since all vertices in I' have an identical neighborhood in G and G', the induction hypothesis implies that every vertex from $I' \cap \mathcal{A}^{i-1}_{G,\mathrm{thr}}(S)$ is in $I' \cap \mathcal{A}^{i-1}_{G',\mathrm{thr}'}(S')$. Moreover, since $|I'(i-1)| < t_{\max}$, we

know that $I(i-1) \setminus S = I'(i-1) \setminus S'$ and since for every vertex $v \in S \cap I$ there is one vertex in $S' \cap I'$, we conclude that $|I'(i-1)| = |I(i-1)|$.

For the running time note the following. A critical independent set I with $|I| > t_{\max}$ can be found by computing all critical independent sets of G in $O(n+m)$ time (McConnell and Spinrad, 1994). It is not hard to see that sorting the vertices in I ensures that the total running time of one application Reduction Rule 4 is bounded by $O(n+m)$. □

Theorem 8 Target Set Selection *admits a problem kernel with at most* $O(2^\tau \cdot t_{\max})$ *vertices, where* τ *and* $t_{\max}$ *denote the vertex cover number and the maximum threshold value of the input graph, respectively. The kernelization runs in* $O(n \cdot (n+m))$ *time.*

Proof Let $G = (V, E)$ denote a graph that is reduced with respect to Reduction Rule 1 and Reduction Rule 4. We show that $|V| \leq \tau + 2^\tau \cdot t_{\max}$. To this end, let Z denote a size-τ vertex cover for G. Note that the vertices in $V \setminus Z$ are contained in at most 2^τ critical independent sets (one for every subset of Z). Moreover, since G is reduced with respect to Reduction Rule 4 (which requires G to be reduced with respect to Reduction Rule 1) the number of vertices in each of these critical independent sets is bounded by $t_{\max}$, and, hence, $|V \setminus Z| \leq 2^\tau \cdot t_{\max}$. Thus, $|V| = |V \setminus Z| + |Z| \leq 2^\tau \cdot t_{\max} + \tau$.

For the running time note that, since each application of Reduction Rule 4 reduces the number of vertices by at least one, by Lemma 10, an instance reduced with respect to Reduction Rule 4 can be computed in $O(n \cdot (n+m))$ time. □

The given problem kernel size is exponential in the parameter vertex cover number. Since every problem that is fixed-parameter tractable has a problem kernel (not necessarily of polynomial size), it is natural to ask whether there is a kernel with size polynomial in the parameter (Bodlaender, 2009; Guo and Niedermeier, 2007). In the following, we answer this question negatively under reasonable complexity-theoretic assumptions.

Dom et al (2009) showed that Hitting Set with universe U and hitting set size k' does not admit a problem kernel of size $(|U| + k')^{O(1)}$ unless an unexpected complexity-theoretic collapse occurs. Bodlaender et al (2011b) introduced a refined concept of parameterized reduction (called polynomial time and parameter transformation) that allows to transfer such hardness results to new problems. By a reduction from Hitting Set to TSS, we can show that, under reasonable complexity-theoretic assumptions, TSS does not admit a problem kernel of size $(\tau + k)^{O(1)}$.

We use the concept of polynomial time and parameter transformations (Bodlaender et al, 2011b). Basically, a parameterized problem A is *polynomial time and parameter reducible* to a parameterized problem B if there exists a parameterized reduction from A to B that runs in polynomial time and ensures that the parameter of the constructed instance of B is polynomial in the parameter of the corresponding instance of A. See Bodlaender et al (2011b, Definition 7) for a formal definition. Bodlaender et al (2011b) have shown that, if the unparameterized versions of A and B are NP-complete, then a polynomial problem kernel for B implies a polynomial problem kernel for A.

We show that Hitting Set parameterized by $|U|$ and k' is polynomial time and parameter reducible to Target Set Selection parameterized by the vertex

cover number τ and target set size k. As a consequence, we arrive at the following theorem.

Theorem 9 Target Set Selection *does not admit a problem kernel of size* $(\tau + k)^{O(1)}$ *unless* coNP $\subseteq$ NP/poly*, where* k *and* τ *denote the target set size and the vertex cover number of the input graph, respectively. This result even holds for bipartite graphs with majority thresholds.*

Proof We show that Hitting Set, parameterized by the combined parameter size of the universe $|U|$ and target set size k', is polynomial time and parameter reducible to Target Set Selection on bipartite graphs with majority thresholds, parameterized by the combined parameter "vertex cover number τ and target set size k". As a consequence, since (the unparameterized versions of) TSS and Hitting Set are NP-complete, TSS on bipartite graphs with majority thresholds does not admit a problem kernel of size $(\tau+k)^{O(1)}$ unless coNP $\subseteq$ NP/poly (Dom et al, 2009).

Let $(\mathcal{F}, U, k')$ denote a Hitting Set-instance consisting of a set family $\mathcal{F} = \{F_1, \ldots, F_m\}$ over a universe $U = \{u_1, \ldots, u_n\}$ and an integer $k' \geq 0$. Recall the definition of the incidence graph $G(U, \mathcal{F}) = (V_U \cup W_\mathcal{F}, E)$ (see Section 3.1). By Lemma 1, setting the threshold of every vertex in $v \in V_U$ to the degree of v and for every vertex $w \in W_\mathcal{F}$ to one, we obtain that Hitting Set is polynomial time and parameter reducible to TSS on bipartite graphs with the combined parameter "vertex cover number and target set size" (note that, since $G(U, \mathcal{F})$ is bipartite, V_U is a vertex cover of $G(U, \mathcal{F})$ of size $|U|$).

In the following, we refine this reduction to work for majority thresholds. To this end, we construct a TSS-instance $(G = (V, F), \mathrm{thr}, k)$. For ease of presentation, let $\Gamma := G(U, \mathcal{F})$ and $h(x) := \deg_\Gamma(x)$ for $x \in V_U \cup W_\mathcal{F}$. The graph G is constructed in several steps from Γ.

First, we set $G := \Gamma$. Then, for every element-vertex $v \in V_U$ we add $h(v)$ dummy vertices $d_v^1, \ldots, d_v^{h(v)}$ and the edges $\{v, d_v^i\}$, $1 \leq i \leq h(v)$. This ensures that $\deg_G(v)/2$ equals the degree of v in Γ, that is, $\deg_G(v)/2 = \deg_\Gamma(v)$.

Second, we add a set $X_U := \{x_1, \ldots, x_{|U|-1}\}$ of $|U| - 1$ vertices to G. These vertices are connected to vertices in $W_\mathcal{F}$ as follows. For each vertex $w \in W_\mathcal{F}$ we add the edges $\{w, x_i\}$ for all $1 \leq i \leq h(w) - 1$. This construction ensures that, assuming majority thresholds, a vertex $w \in W_\mathcal{F}$ becomes active if all vertices in X_U and one vertex $v \in V_U$ are active before w. Let H' denote the graph constructed so far.

Third, we enforce that the vertices in X_U are contained in every minimal target set. To this end, for every vertex $x_i \in X_U$ we add $\deg_{H'}(x_i) + 1$ dummy vertices $d_{x_i}^1, \ldots, d_{x_i}^{\deg_{H'}(x_i)+1}$ and the edges $\{x_i, d_{x_i}^j\}$ for all $1 \leq j \leq \deg_{H'}(x_i) + 1$. This completes the construction of G. Finally, set $\mathrm{thr}(v) := \deg_G(v)/2$ (majority thresholds) for all $v \in V(G)$. This completes the construction.

Let D denote the set of all dummy vertices. Note that G is bipartite; one partition contains the vertices $V_1 := V_U \cup X_U$, the other partition contains the vertices $V_2 := W_\mathcal{F} \cup D$, and there are only edges between V_1 and V_2. Moreover, since G is bipartite, V_1 is a vertex cover of G of size $2|U| - 1$.

For the correctness of the reduction, we show the that $(\mathcal{F}, U, k')$ is a yes-instance of HS if and only if$(G, \mathrm{thr}, k' + |X_U|)$ is a yes-instance of TSS.
"$\Rightarrow$": Let $U' \subseteq U$ denote a hitting set. It is not hard to verify that $S := \{v_u : u \in U'\} \cup X_U$ is a target set for G.

"⇐": Let S be a target set for G of size at most $|X_U| + k'$. First, we show that $X_U \subseteq S$. By Observation 2 we can assume that S does not contain any vertex with threshold one. In particular, $S \cap D = \emptyset$. Consider an arbitrary vertex $x \in X_U$. Since $|N(x) \setminus D| < \mathrm{thr}(x)$, it holds that $x \in S$. Let $S' := S \setminus X_U$ and let (G', thr', k') denote the X_U-reduced instance of $(G, \mathrm{thr}, k' + |X_U|)$. By Observation 5, S' is a target set for G' of size at most k'. Moreover, for a vertex $w \in W_{\mathcal{F}}$, we have $\mathrm{thr}'(w) = 1$ since $\mathrm{thr}(w) = h(x)$ and $|N_G(w) \cap X_U| = h(x) - 1$. Hence, by Observation 2 we can assume that $S' \subseteq V_U$. Furthermore, observe that $\mathrm{thr}(v) = \mathrm{thr}'(v) = h(v)$ for all vertices $v \in V_U$. Hence, by the same arguments as in the proof of Lemma 1 (see Section 3.1), it follows that $U' := \{u : v_u \in S'\}$ is a hitting set of size at most k. □

6 Parameterization by the Feedback Edge Set Number

A further structural parameter of a graph G that is lower-bounded by the treewidth of G is the size f of a smallest feedback edge set of G, that is, a smallest set of edges of G whose deletion makes G acyclic. Recall that TSS is $W[1]$-hard with respect to the parameter feedback vertex set (Ben-Zwi et al, 2011). Some real-world social networks are tree-like (Chen, 2009) (for example, sexual networks (Potterat et al, 2002)) or scale-free (Franks et al, 2008). This often corresponds to small feedback edge sets and motivates parameterizing TSS with f. Like the vertex cover number τ, the feedback edge set number f is a "weaker" parameter than treewidth, raising hope for faster algorithms for TSS in cases where f is small. See also Uhlmann and Weller (2010) for another problem studied with parameter feedback edge set. A clear advantage over treewidth (and tree decompositions) is that an optimal feedback edge set can be determined efficiently by computing a spanning tree.

In the following, we first show a problem kernel of size $O(f)$ for TSS based on two data reduction rules and then present an algorithm that solves TSS in $O(4^f \cdot f + m)$ time. Both results hold for arbitrarily thresholds and, hence, we do not look at special threshold functions in this section.

6.1 Kernelization

Here, we present two data reduction rules for TSS that can be applied exhaustively in linear time and leave a problem kernel of size $O(f)$. Reduction Rule 5 removes vertices v with $\mathrm{thr}(v) = \deg(v) = 1$ from G. Note that Reduction Rule 1 (see Section 2) removes all other degree-one vertices from G.

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.*

The correctness of Reduction Rule 5 follows immediately from Observation 2 (Lemming Vertices) in Section 2. One easily verifies that Reduction Rule 5 can be exhaustively applied in linear time. In a graph reduced with respect to Reduction Rules 1 and 5, we can bound the number of vertices with degree at least three by $2f$.

Lemma 11 *A graph with feedback edge set number f reduced with respect to Reduction Rules 1 and 5 has at most $2f$ vertices with degree at least three.*

Proof Consider a graph $G = (V, E)$ that is reduced with respect to Reduction Rules 1 and 5 and a minimum feedback edge set F of G. Let V_1,V_2, and $V_{\geq 3}$ denote the sets of vertices of G whose degree in $G - F$ is one, two, and at least three, respectively. Since $G-F$ is a forest, we have $|V_{\geq 3}| \leq |V_1|$. Let V_2' be the set of vertices of $V_1 \cup V_2$ that have degree at least three in G. Note that $2|F| \geq |V_1| + |V_2'|$. We can now upper-bound the number of vertices of degree at least three in G by $|V_{\geq 3}| + |V_2'| \leq 2|F| = 2f$. □

With Lemma 11 we upper-bound the number of vertices with degree at least three in graphs that are reduced with respect to Reduction Rules 1 and 5. Since these graphs do not contain degree-one vertices, it remains to bound the degree-two vertices.

In the following, *bypassing* a vertex $v \in V$ with $N(v) = \{u, w\}$ means to delete v from G and to insert the edge $\{u, w\}$. Note that, if u and w are already neighbors in G, then v must not be bypassed.

Reduction Rule 6 *Let (G, thr, k) be an instance of* TSS *reduced with respect to Reduction Rule 1 and let (u, v, w) be a path in G with $\deg(u) = \deg(v) = \deg(w) = 2$ where u and w are neither neighbors nor twins. If there is a vertex $x \in \{u, v, w\}$ with $\mathrm{thr}(x) = 1$, then bypass x. Otherwise, bypass u and w and decrease k by one.*

Lemma 12 *Reduction Rule 6 is correct and can be applied exhaustively in linear time.*

Proof Let $I := (G, \mathrm{thr}, k)$ be an instance of TSS and let G contain a path (u, v, w) as described in Reduction Rule 6. Let $I' := (G' = (V', E'), \mathrm{thr}', k')$ denote the result of applying Reduction Rule 6 to (u, v, w). Note that u and w being degree-two vertices that are neither neighbors nor twins implies $N_G[u] \cap N_G[w] = \{v\}$ and, hence, bypassing u and w is possible. If there is a vertex $x \in \{u, v, w\}$ with $\mathrm{thr}(x) = 1$, then the correctness follows immediately from Observation 6 (Subdivision) in Section 2. Thus, we assume that $\mathrm{thr}(u) = \mathrm{thr}(v) = \mathrm{thr}(w) = 2$ (recall that I is reduced with respect to Reduction Rule 1). We show that G has a target set of size k if and only if G' has a target set of size $k' := k - 1$.

"⇒": Let S denote a target set of size k for G. If $|S \cap \{u, v, w\}| = 1$, then $S \cap \{u, v, w\} = \{v\}$ because, otherwise, only one of u and w can be activated, implying $v \notin \mathcal{A}_{G', \mathrm{thr}'}(S)$. Then, $S' := S \setminus \{v\}$ is a target set for $G - \{u, v, w\} = G' - \{v\}$. By Observation 4 (Vertex Addition), S' is also a target set for G'. If $|S \cap \{u, v, w\}| \geq 2$, then we can assume that $u, w \in S$. Let I^* denote the $\{u, w\}$-reduced instance of I. It is not hard to see that I^* equals the $\{v\}$-reduced instance of I'. Then, by Observation 5 (Chain Reduction), $S' := (S \cup \{v\}) \setminus \{u, w\}$ is a target set for G'.

"⇐": Let S' denote a target set of size $k - 1$ for G'. If $v \in S'$, then the $\{v\}$-reduced instance of I' equals the $\{u, w\}$-reduced instance of I. Then, by Observation 5, $S := (S' \setminus \{v\}) \cup \{u, w\}$ is a target set for G. If $v \notin S'$, then S' is also a target set for $G' - \{v\} = G - \{u, v, w\}$. Let $(G^*, \mathrm{thr}^*, k^*)$ denote the S'-reduced instance of I. Then, G^* consists of the path (u, v, w) with $\mathrm{thr}^*(u) = \mathrm{thr}^*(w) = 1$ and $\mathrm{thr}^*(v) = 2$ and $k^* = 1$. Then, $\{v\}$ is a target set for G^* and, by Observation 5, $S' \cup \{v\}$ is a target set for G.

By precomputing all twins in linear time (McConnell and Spinrad, 1994) and maintaining a list of adjacent degree-two vertices for each vertex, we can apply Reduction Rule 6 in linear time. □

The presented data reduction rules are enough to produce a $12f$-vertex kernel that can be computed in linear time.

Theorem 10 Target Set Selection *admits a problem kernel of size $O(f)$, where f denotes the feedback edge set number. The kernelization runs in linear time.*

Proof Let (G, thr, k) be an instance of TSS reduced with respect to Reduction Rules 1, 5, and 6. Furthermore, let F denote an optimal feedback edge set for G. Thus, $T := G - F$ is a forest. Consider the graph T^* that results from bypassing all vertices in T having degree two in both G and T. It is easy to see that all vertices in T^* have the same degree in T^* as they have in T. Hence, each vertex v with degree two in T^* is incident to an edge of F in G because otherwise it would have been bypassed. Furthermore, each leaf in T^* not incident to an edge of F is also a leaf in G, but since G is reduced with respect to Reduction Rules 1 and 5, each leaf of T^* is incident to an edge of F. Consequently, the number of vertices of T^* with degree at most two is bounded by $2f$. Furthermore, by Lemma 11, the number of vertices of T^* with degree at least three is bounded by $2f$. Thus, the overall number of vertices in T^* is at most $4f$. Finally, since G is reduced with respect to Reduction Rule 6, for each edge $\{u, v\}$ of T^*, there are at most two vertices between u and v in T. Hence, the overall number of vertices of T can be bounded by $12f$ and since T is a forest, so can the overall number of edges of T. By construction of T, we can bound the overall number of vertices of G by $12f$ and the overall number of edges of G by $13f$.

By Lemma 12, any graph can be reduced with respect to Reduction Rules 1, 5, and 6 in linear time. □

6.2 Fixed-Parameter Algorithm

Theorem 10 already implies that TSS is fixed-parameter tractable by applying a brute-force search to the size-$O(f)$ problem kernel. Here, we develop a more efficient fixed-parameter algorithm.

Our algorithm uses a search tree to compute a solution for an input instance (G, thr, k) for TSS if there is one. The algorithm uses two types of branchings, one branches on degree-two vertices that are in a cycle, the other branches on degree-at-least-three vertices.

In the algorithm, we use the notion of *branching rules*. Here, we analyze an instance I and replace it with a set $\mathcal{I}$ of new instances. The creation of new instances is defined in branching rules, which we call *correct* if the original instance I is a yes-instance if and only if there is a yes-instance in $\mathcal{I}$. The maximum number of instances created in such a step is called the *branching number* of the corresponding branching rule. In analogy to data reduction rules, instances that are not subject to a specific branching rule $\mathcal{R}$ are called *reduced* with respect to $\mathcal{R}$. By considering each application of a branching rule as a vertex labeled I with the elements of $\mathcal{I}$ as children, we can define a *search tree* for each input instance.

In the following, $V_{\bigcirc}(G)$ denotes the set of all vertices of a graph G that are in cycles. Let C be some two-edge-connected component of G containing at least three vertices, that is, C is a non-singleton connected component that does not contain bridges[9] of G.

[9] A bridge of a graph G is an edge whose deletion disconnects G.

Branching on degree-two vertices. The first branching rule branches on vertices in $V_{\bigcirc}(G)$ with degree two and threshold two. To this end, we present a branching rule with branching number two that decreases the feedback edge set number f in every application.

Branching Rule 1 *Let $I := (G = (V, E), \mathrm{thr}, k)$ be an instance of* TSS *and let $v \in V_{\bigcirc}(G)$ and $\mathrm{thr}(v) = \deg(v) = 2$. Then, create the $\{v\}$-reduced instance of I (see Definition 1) and create the instance $(G - v, \mathrm{thr}, k)$.*

Since $f = m - (n - \#\mathrm{cc})$ where $\#\mathrm{cc}$ is the number of connected components and removing an edge from a cycle increases neither $\#\mathrm{cc}$ nor n, it is clear that f decreases by one as stated by the next observation.

Observation 9 *Let $G = (V, E)$ be some graph, and let $e \in E$ be an edge of G that is in some cycle of G. Then deleting e decreases the feedback edge set number of G.*

With Observation 9, we can now prove the correctness of Branching Rule 1.

Lemma 13 *Branching Rule 1 is correct and each application decreases the feedback edge set number of the input graph. Each application of Branching Rule 1 can be performed in linear time.*

Proof Let $I = (G, \mathrm{thr}, k)$ denote the input instance and let S be an optimal target set for G (note that I is a yes-instance if and only if $|S| \leq k$). If $v \in S$, then since $\mathrm{thr}(v) = \deg(v)$, by Observation 5, reducing I with respect to $\{v\}$ produces an instance that is equivalent to I. If $v \notin S$, then, by Observation 4, $(G - v, \mathrm{thr}, k)$ is equivalent to (G, thr, k). In both cases, v is deleted and since $v \in V_{\bigcirc}(G)$, at least one of the edges that are incident to v is in a cycle. By Observation 9, the feedback edge set number of both created instances is smaller than the feedback edge set number of G.

For the running time, note that the vertices that are in cycles in G can be found by computing all bridges of G, which can be done in linear time (Tarjan, 1974). Furthermore, the $\{v\}$-reduced instance of I can be computed in linear time (see running time of Reduction Rule 1, Section 2), implying that an application of Branching Rule 1 can be performed in linear time. □

It is easy to see that Branching Rule 1 branches into at most two cases. After each application of Branching Rule 1, our algorithm exhaustively applies Reduction Rules 1 and 5 to the created instances. If none of Branching Rule 1 and Reduction Rules 1 and 5 can be applied to the graph, we employ a second branching rule that branches on each degree-at-most-three vertex of $V_{\bigcirc}(G)$.

Branching on degree-at-least-three vertices. The second branching rule deals with vertices with degree at least three that are contained in cycles. To present the branching rule, we introduce the following notation. Recall that the set of degree-two vertices of the input graph G is denoted by V_2 and the vertices that are contained in cycles in G is denoted by $V_{\bigcirc}(G)$. For a two-edge-connected component C of G, we define $V_2(C) = V_{\bigcirc}(C) \cap V_2$. Note that, if each two-edge-connected component of G is contracted into a single vertex, then the remaining graph is a forest. In the following, we call this forest the *component forest* T of G. The component forest of a graph can be computed in linear time by removing all bridges and computing the remaining connected components.

We will apply the second branching rule only to instances that are not subject to any of Reduction Rules 1 and 5 and Branching Rule 1. Note that, in this case, every leaf of T corresponds to a contracted non-singleton two-edge-connected component of G.

Branching Rule 2 *Let $I := (G, \mathrm{thr}, k)$ be an instance of* TSS *that is reduced with respect to Reduction Rules 1 and 5 and Branching Rule 1. Let T be the component forest of G and let C denote a two-edge-connected component of G corresponding to a leaf in T. Finally, let v be the unique vertex in $N_G(V(C))$, and let w be the unique vertex in $N(v) \cap V(C)$. Then, for each $V' \subseteq V(C) \setminus V_2$, create a new instance by modifying I in the following way:*

If V' is a target set for C,
then create the V'-reduced instance of I.
Else,
if V' is a target set for C with $\mathrm{thr}(w)$ decreased by one,
then delete $V(C)$ from G and decrease k by $|V'|$.
Otherwise, create a trivial no-instance.

Lemma 14 *Branching Rule 2 is correct and the relevant vertices can be computed in linear time.*

Proof We show that I is a yes-instance if and only if at least one of the created instances is a yes-instance.

"⇒": Let S be a solution for I and let $S_C := S \cap V(C)$. Since I is reduced with respect to Reduction Rule 1 and Branching Rule 1, all vertices in $V_2(C)$ have threshold one. Thus, by Observation 2, there is an optimal solution for I that does not contain vertices of V_2, implying $S_C \subseteq V(C) \setminus V_2$. If S_C is a target set for C, then there is a $V' = S_C$ such that the V'-reduced instance of I is created by Branching Rule 2 and is a yes-instance. Otherwise, since S is a solution for I, the set $S \setminus S_C$ is a target set for $G - V(C)$ and, hence, the instance $(G - V(C), \mathrm{thr}, k - |V'|)$, which is created by Branching Rule 2, is a yes-instance.

"⇐": Let I' be a yes-instance that is created by Branching Rule 2 and let S' be a solution for I'. Let V' be the subset of $V(C)$ that corresponds to I'. If V' is a target set for C, then, by Observation 5, $S' \cup V'$ is a solution for I. Otherwise, V' is a target set for C with $\mathrm{thr}(w)$ decreased by one. However, since v is activated by S', in the S-reduced instance of I, the threshold of w is decreased by one. Hence, by Observation 5, $S' \cup V'$ is a solution for I.

For the running time, note that we can apply Branching Rule 2 by computing the component forest T of G, finding a leaf of T and listing all vertices of degree at least three in the two-edge-connected component corresponding to this leaf. □

To analyze the branching number of Branching Rule 2, let C, v, and w be defined as in Branching Rule 2. Let G' denote the graph that results from removing the edge $\{v, w\}$ from G and let C' denote the connected component of G' that contains all vertices of C. Let f_C denote the feedback edge set of C'. Note that $f_C > 0$ because the input is reduced with respect to Reduction Rules 1 and 5. Since Lemma 11 applies to C', the number of vertices in $V(C) \setminus V_2$ is at most $2f_C + 1$ (mind that w may have degree-three in G but only degree-two in G'). Since C is completely erased in each of the instances created by Branching Rule 2, the feedback edge set number decreases by f_C while, in total, $2^{2f_C} = 4^{f_C}$ instances

are created. For the sake of analyzing the search tree size, we can thus assume each application of Branching Rule 2 to decrease the feedback edge set number by one while creating four instances. This way, Branching Rule 2 can be pictured as, for two vertices x, y of $V_{\bigcirc}(G) \setminus V_2$, branching into the four possibilities of including x or y (or none of them) into the solution.

Theorem 11 Target Set Selection *can be solved in* $O(4^f \cdot f + m)$ *time, where* f *denotes the feedback edge set number of the input graph.*

Proof By Lemma 13, Branching Rule 1 decreases the feedback edge set number. Also since Branching Rule 1 creates only two instances, its branching number is two.

As argued above, we picture Branching Rule 2 as a branching rule with a branching number of four that decreases the feedback edge set number. Clearly, neither Reduction Rule 1 nor Reduction Rule 5 increase the feedback edge set number of the graph. We can thus bound the worst-case search tree size by 4^f with f denoting the feedback edge set number of the input graph.

In each search tree node, we apply Reduction Rules 1 and 5. These reduction rules can be applied exhaustively in linear time. Executing the branching rules takes linear time per search tree node. Since an instance that is not subject to any of the branching and reduction rules is a trivial yes-instance or a trivial no-instance, the whole algorithms runs in $O(4^f \cdot (n + m))$ time. By first applying the linear size problem kernel stated in Theorem 10, the running time $O(4^f \cdot f + m)$ can be achieved.

The correctness of the algorithm follows from the correctness of the employed data reduction and branching rules. □

7 Conclusion and Future Challenges

Following the spirit of multivariate algorithmics (Fellows, 2009; Niedermeier, 2010), we studied several natural parameterizations of TSS. We confirmed the intuition deriving from the hardness of Dominating Set that TSS remains hard for the parameter diameter. However, for the structural parameters "cluster editing number", "vertex cover number", and "feedback edge set number", we established tractability results by providing fixed-parameter algorithms and kernelizations. Table 1 in Section 1 summarizes our results on the complexity of TSS for these restricted cases. In this context, it is notable that our algorithms for the parameters "cluster editing number" and "vertex cover number" are merely brute-force algorithms on the respective (weak) kernels. It remains to develop faster algorithms possibly exploiting these structural parameters better than our kernelizations.

There are numerous challenges for future work on parameterized and multivariate complexity analysis of TSS, both in terms of new parameters and in terms of parameter combinations. In particular, since social networks mostly have small diameters and since we have shown that parameterizing by the diameter alone is hopeless for achieving fixed-parameter tractability, a natural next step is to combine other parameters with the diameter and to study parameterized complexity using these combined parameters. Another line of future research on Target Set Selection could be to further extend the problem analysis by using

a more fine-grained classification of threshold functions and their impact on problem complexity. In addition, since for several applications the parameter "number of activation rounds" appears to be relevant and should be small, this motivates to investigate the influence of this parameter on the computational complexity of TSS. Since TSS with only one round is an already NP-hard dominating set variant, this calls, similarly to diameter, for the combination with other parameters. Moreover, note that we focused on activating *all* graph vertices; it is natural to extend our studies to cases when only a specified fraction of the graph vertices shall be activated at minimum cost. Indeed, for applications related to viral marketing or influence spreading this actually seems to be the scenario which occurs more often than trying to activate all social network vertices. Clearly, our work so far is of purely theoretical nature and our fixed-parameter tractability results call for an empirical evaluation. Finally, it remains a task of future study to exploit further parameterizations of (social) networks that make TSS tractable in relevant special cases.

References

Abrahamson KA, Downey RG, Fellows MR (1995) Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. Annals of Pure and Applied Logic 73(3):235–276 3

Ackerman E, Ben-Zwi O, Wolfovitz G (2010) Combinatorial model and bounds for target set selection. Theoretical Computer Science 411(44-46):4017–4022 3

Agarwal N, Liu H, Tang L, Yu P (2011) Modeling blogger influence in a community. Social Network Analysis and Mining To appear 3

Bazgan C, Chopin M, Fellows MR (2011) Parameterized complexity of the firefighter problem. In: Proc. 22nd ISAAC, Springer, LNCS, vol 7074, pp 643–652 4

Ben-Zwi O, Hermelin D, Lokshtanov D, Newman I (2011) Treewidth governs the complexity of target set selection. Discrete Optimization 8(1):87–96 3, 4, 5, 25, 30

Böcker S (2011) A golden ratio parameterized algorithm for cluster editing. In: Proc. 22nd IWOCA, Springer, LNCS, vol 7056, pp 85–95 22

Böcker S, Damaschke P (2011) Even faster parameterized cluster deletion and cluster editing. Information Processing Letters 111(14):717 – 721 23

Bodlaender HL (2009) Kernelization: New upper and lower bound techniques. In: Proc. 4th IWPEC, Springer, LNCS, vol 5917, pp 17–37 8, 28

Bodlaender HL, Jansen BMP, Kratsch S (2011a) Preprocessing for treewidth: A combinatorial analysis through kernelization. In: Proc. 38th ICALP, Springer, LNCS, vol 6755, pp 437–448 5, 25

Bodlaender HL, Thomassé S, Yeo A (2011b) Kernel bounds for disjoint cycles and disjoint paths. Theoretical Computer Science 412(35):4570–4578 28

Brandstädt A, Le VB, Spinrad JP (1999) Graph Classes: a Survey, SIAM Monographs on Discrete Mathematics and Applications, vol 3. SIAM 7

Chen J, Kanj IA, Xia G (2010) Improved upper bounds for vertex cover. Theoretical Computer Science 411(40-42):3736–3756 26

Chen N (2009) On the approximability of influence in social networks. SIAM Journal on Discrete Mathematics 23(3):1400–1415 3, 6, 8, 30

Cygan M, Fomin F, Leeuwen EJV (2012) Parameterized complexity of firefighting revisited. In: Proc. 6th IPEC, Springer, LNCS, vol 7112, pp 13–26 4

Davidson SB, Garcia-Molina H, Skeen D (1985) Consistency in partitioned networks. ACM Computing Surveys 17(3):341–370 2

De Nooy W, Mrvar A, Batagelj V (2011) Exploratory Social Network Analysis with Pajek. No. 34 in Structural Analysis in the Social Sciences, Cambridge University Press 6

Dom M, Lokshtanov D, Saurabh S (2009) Incompressibility through colors and IDs. In: Proc. 36th ICALP, Springer, LNCS, vol 5555, pp 378–389 28, 29

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proc. 7th ACM KDD, ACM Press, pp 57–66 3

Downey R, Fellows M, McCartin C, Rosamond F (2008) Parameterized approximation of dominating set problems. Information Processing Letters 109(1):68–70 5, 11

Downey RG, Fellows MR (1999) Parameterized Complexity. Springer 8, 9, 11, 12

Dreyer PA Jr, Roberts FS (2009) Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. Discrete Applied Mathematics 157:1615–1627 2, 3

Easley D, Kleinberg J (2010) Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press 4, 6

Fellows M (2009) Towards fully multivariate algorithmics: Some new results and directions in parameter ecology. In: Proc. 20th IWOCA, Springer, LNCS, vol 5874, pp 2–10 4, 35

Fellows MR, Lokshtanov D, Misra N, Rosamond FA, Saurabh S (2008) Graph layout problems parameterized by vertex cover. In: Proc. 19th ISAAC, Springer, LNCS, vol 5369, pp 294–305 5, 25

Fiala J, Golovach PA, Kratochvíl J (2011) Parameterized complexity of coloring problems: Treewidth versus vertex cover. Theoretical Computer Science 412(23):2513–2523 5, 25

Flum J, Grohe M (2006) Parameterized Complexity Theory. Springer 8, 9

Franks DW, Noble J, Kaufmann P, Stagl S (2008) Extremism propagation in social networks with hubs. Adaptive Behavior 16(4):264–274 6, 30

Garcia-Molina H, Barbará D (1985) How to assign votes in a distributed system. Journal of the ACM 32(4):841–860 2

Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman 11

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12):7821–7826 4, 5, 11

Guo J, Niedermeier R (2007) Invitation to data reduction and problem kernelization. ACM SIGACT News 38(1):31–45 8, 28

Harant J, Pruchnewski A, Voigt M (1999) On dominating sets and independent sets of graphs. Combinatorics, Probability and Computing 8(6):547–553 3

Jiang H, Zhang C, Zhu B (2010) Weak kernels. Tech. Rep. TR10-005, Department of Computer Science, Montana State University 21

Johnson DS (1984) The genealogy of theoretical computer science: a preliminary report. SIGACT News 16(2):36–49, reprinted in Bulletin of the EATCS, No. 25, pp. 198-211, 1985. 6

Jurvetson S (2000) What exactly is viral marketing? Red Herring 78:110–112 2

Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proc. 9th ACM KDD, ACM Press, pp 137–146 2, 3

Kutten S, Peleg D (1999) Fault-local distributed mending. Journal of Algorithms 30(1):144–165 2

Leskovec J, Horvitz E (2008) Planetary-scale views on a large instant-messaging network. In: Proc. 17th WWW, ACM Press, pp 915–924 4

Leskovec J, Adamic LA, Huberman BA (2007a) The dynamics of viral marketing. ACM Transactions on the Web 1(1) 2

Leskovec J, McGlohon M, Faloutsos C, Glance NS, Hurst M (2007b) Patterns of cascading behavior in large blog graphs. In: Proc. 7th SDM, SIAM, pp 551–556 2, 4

MacGillivray G, Wang P (2003) On the firefighter problem. Journal of Combinatorial Mathematics and Combinatorial Computing 47:83–96 4

McConnell RM, Spinrad J (1994) Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In: Proc. 5th SODA, ACM/SIAM, pp 536–545 19, 24, 26, 28, 31

Milgram S (1967) The small world problem. Psychology Today 1:61–67 4

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45(2):167–256 3, 4, 5, 11

Newman MEJ (2010) Networks: An Introduction. Oxford University Press 3, 4, 5, 6, 11

Newman MEJ, Park J (2003) Why social networks are different from other types of networks. Physical Review E 68(3):036,122 4, 5, 11

Niedermeier R (2006) Invitation to Fixed-Parameter Algorithms. Oxford University Press 8, 9

Niedermeier R (2010) Reflections on multivariate algorithmics and problem parameterization. In: Proc. 27th STACS, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, LIPIcs, vol 5, pp 17–32 4, 35

Norlen K, Lucas G, Gebbie M, Chuang J (2002) Eva: Extraction, visualization and analysis of the telecommunications and media ownership network. In: Proc. 14th ITS 6

Pelc A (1996) Efficient fault location with small risk. In: SIROCCO, Carleton Scientific, pp 292–300 2

Peleg D (2002) Local majorities, coalitions and monopolies in graphs: a review. Theoretical Computer Science 282:231–257 2, 4

Potterat JJ, Phillips-Plummer L, Muth SQ, Rothenberg RB, Woodhouse DE, Maldonado-Long TS, Zimmerman HP, Muth JB (2002) Risk network structure in the early epidemic phase of HIV transmission in Colorado Springs. Sexually Transmitted Infections 78:159–163 5, 30

Raeder T, Chawla N (2011) Market basket analysis with networks. Social Network Analysis and Mining 1:97–113 3

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proc. 8th ACM KDD, ACM Press, pp 61–70 3

Shamir R, Sharan R, Tsur D (2004) Cluster graph modification problems. Discrete Applied Mathematics 144(1–2):173–182 18

Tarjan RE (1974) A note on finding the bridges of a graph. Information Processing Letters 2(6):160–161 33

Uhlmann J, Weller M (2010) Two-layer planarization parameterized by feedback edge set. In: Proc. 7th TAMC, Springer, LNCS, vol 6108, p 431–442 30

Wang P, Moeller SA (2002) Fire control on graphs. Journal of Combinatorial Mathematics and Combinatorial Computing 41:19–34 4

Watts D, Peretti J, Frumin M (2007) Viral marketing for the real world. Harvard Business Review 85(5):22–23 2

Zhang W, Wu W, Wang F, Xu K (2012) Positive influence dominating sets in power-law graphs. Social Network Analysis and Mining 2(1):31–37 3