

Target Set Selection in Dense Graph Classes

Pavel Dvořák¹

Computer Science Institute, Charles University, Prague, Czech Republic
koblich@iuuk.mff.cuni.cz

Dušan Knop²

Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany *and*
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic
dusan.knop@gmail.com

Tomáš Toufar

Computer Science Institute, Charles University, Prague, Czech Republic
toufi@iuuk.mff.cuni.cz

Abstract

In this paper we study the TARGET SET SELECTION problem from a parameterized complexity perspective. Here for a given graph and a threshold for each vertex the task is to find a set of vertices (called a target set) to activate at the beginning which activates the whole graph during the following iterative process. A vertex outside the active set becomes active if the number of so far activated vertices in its neighborhood is at least its threshold.

We give two parameterized algorithms for a special case where each vertex has the threshold set to the half of its neighbors (the so called MAJORITY TARGET SET SELECTION problem) for parameterizations by the neighborhood diversity and the twin cover number of the input graph.

We complement these results from the negative side. We give a hardness proof for the MAJORITY TARGET SET SELECTION problem when parameterized by (a restriction of) the modular-width – a natural generalization of both previous structural parameters. We show that the TARGET SET SELECTION problem parameterized by the neighborhood diversity when there is no restriction on the thresholds is $W[1]$ -hard.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases parameterized complexity, target set selection, dense graphs

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.18

Related Version arXiv:1610.07530

1 Introduction

We study the TARGET SET SELECTION problem (also called DYNAMIC MONOPOLIES), using notation according to Kempe et al. [16], from parameterized complexity perspective. We use standard notions of parameterized complexity, see [9]. Let $G = (V, E)$ be a graph, $S \subseteq V$, and $f: V \rightarrow \mathbb{N}$ be a *threshold function*. The *activation process* arising from the set $S_0 = S$ is an iterative process with resulting sets S_0, S_1, \dots such that for $i \geq 0$

$$S_{i+1} = S_i \cup \{v \in V : |N(v) \cap S_i| \geq f(v)\},$$

¹ The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 616787.

² Part of this work has been done while affiliated with Department of Informatics, University of Bergen, Norway and supported by the project NFR MULTIVAL.



where by $N(v)$ we denote the set of vertices adjacent to v . Note that after at most $n = |V|$ rounds the activation process has to stabilize – that is, $S_n = S_{n+i}$ for all $i > 0$. We say that the set S is a *target set* if $S_n = V$ (for the activation process $S = S_0, \dots, S_n$).

TARGET SET SELECTION

Input: A graph $G = (V, E)$, $f: V \rightarrow \mathbb{N}$, and a positive integer $b \in \mathbb{N}$.

Task: Find a target set $S \subseteq V$ of size at most b or report that there is no such set.

We call the input integer b the *budget*. The problem interpretation and computational complexity clearly may vary depending on the input function f . There are three important settings studied (as we will discuss later) – namely constant, majority, and a general function. If the threshold function f is the majority (i.e., $f(u) = \lceil \deg(u)/2 \rceil$ for every vertex $u \in V$) we denote the problem as MAJORITY TARGET SET SELECTION.

Motivation. The TARGET SET SELECTION problem was introduced by Domingos and Richardson [10] in order to study influence of direct marketing on a social network. It is noted therein that it captures e.g. *viral marketing* [21]. The TARGET SET SELECTION problem is important also from the graph theoretic viewpoint, since it generalizes many well known NP-hard problems on graphs. These problems include

- VERTEX COVER [4] – set $f(v) = \deg(v)$ for all $v \in V$ and
- IRREVERSIBLE k -CONVERSION SET [11], k -NEIGHBORHOOD BOOTSTRAP PERCOLATION [2] – the TARGET SET SELECTION problem with all thresholds fixed to k .

Previous Results. The TARGET SET SELECTION problem received attention of researchers in theoretical computer science in the past years. A general upper bound on the number of selected vertices under majority constraints is $|V|/2$ [1]. The TARGET SET SELECTION problem admits an FPT algorithm when parameterized by the vertex cover number [19]. A $t^{\mathcal{O}(w)}$ poly(n) algorithm is known, where w is the tree-width of the input graph and t is an upper-bound on the threshold function [3], that is, $f(v) \leq t$ for every vertex v . This is essentially optimal, as the TARGET SET SELECTION problem parameterized by the path-width is W[1]-hard for majority [6] and general functions [3]. The TARGET SET SELECTION problem is solvable in linear time on trees [4] and more generally on block-cactus graphs [5]. The optimization variant of the TARGET SET SELECTION problem is hard to approximate [4] within a polylogarithmic factor. For more and less recent results we refer the reader to a survey by Peleg [20]. Cicalese et al. [8, 7], considered versions of the problem in which the number of rounds of the activation process is bounded. For graphs of bounded clique-width, given parameters a, b, ℓ , they gave polynomial-time algorithms to determine whether there exists a target set of size b , such that at least a vertices are activated in at most ℓ rounds. Recently Hartmann [15] gave a single-exponential FPT algorithm for TARGET SET SELECTION parameterized by clique width when all thresholds are bounded by a constant.

Our Results. In this work we generalize some results obtained by Nichterlein et al. [19]. Chopin et al. [6] essentially proved that in sparse graph classes (such as graphs with the bounded tree-width) parameterized complexity of the MAJORITY TARGET SET SELECTION problem is the same as for the TARGET SET SELECTION problem. For these graph classes, it is not hard to see that e.g. if the threshold for vertex v is set above the majority (i.e., $f(v) > \lceil \deg(v)/2 \rceil$), then we may add $2(f(v) - \lceil \deg(v)/2 \rceil)$ vertices neighboring with v only and the parameter stays unchanged whereas the threshold of v dropped to majority. However, this is not true in general for dense graph classes. We demonstrate this phenomenon for the parameterization by the neighborhood diversity. We show parameterized algorithm for

a function which generalizes both constant and majority threshold functions. We call this function uniform (and the corresponding problem UNIFORM TARGET SET SELECTION), see the next section for a proper definition. Roughly speaking all vertices belonging to a same part of a graph decomposition must possess the same value of the threshold function. In a slight contrast to the previous results, we derive an FPT algorithm that, instead of the maximal threshold value t , depends on the size of the image of the threshold function for graphs having bounded neighborhood diversity.

► **Theorem 1.** *There is an FPT algorithm for the UNIFORM TARGET SET SELECTION problem parameterized by the neighborhood diversity of the input graph.*

► **Theorem 2.** *The TARGET SET SELECTION problem is W[1]-hard parameterized by the neighborhood diversity of the input graph.*

The complexity of the MAJORITY TARGET SET SELECTION problem is not resolved for parameterization by the cluster vertex deletion number [6] (the number of vertices whose removal from the graph results in a collection of disjoint cliques). We have a positive result for a slightly stronger parameterization: we assume that for every vertex we remove and every clique the vertex is either completely adjacent to the whole clique or is completely nonadjacent. This result also suggests that various weighted variants of the TARGET SET SELECTION problem may be in FPT when parameterized by the vertex cover number.

► **Theorem 3.** *There is an FPT algorithm for the UNIFORM TARGET SET SELECTION problem parameterized by the size of the twin cover of the input graph.*

Previous results [6] imply that the parameterized complexity of the TARGET SET SELECTION and the MAJORITY TARGET SET SELECTION problems is the same in graphs with bounded clique-width. Of course, much more is known – the proof herein shows that the MAJORITY TARGET SET SELECTION problem is W[1]-hard on graphs of the bounded tree-depth (even though only the tree-width is claimed). We show that this is already the case for parameterization by the (restricted) modular-width that generalizes both neighborhood diversity and twin cover number.

► **Theorem 4.** *The MAJORITY TARGET SET SELECTION problem is W[1]-hard parameterized by the modular-width of the input graph.*

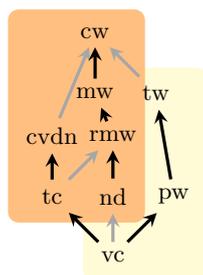
2 Preliminaries on Structural Graph Parameters

We give formal definitions of several graph parameters used in this work. To get better acquainted with these parameters, we provide a map of the considered parameters in Figure 1.

For a graph $G = (V, E)$ the set $U \subseteq V$ is called a *vertex cover* of G if for every edge $e \in E$ it holds that $e \cap U \neq \emptyset$. The *vertex cover number* of a graph, denoted as $vc(G)$, is the least integer k for which there exists a vertex cover of size k .

As the vertex cover number is (usually) too restrictive, many authors focused on defining other (i.e., weaker) structural parameters. Three most well-known parameters of this kind are the path-width, the tree-width, and the clique-width. Classes of graphs with the bounded tree-width (respectively the path-width) are contained in the so called sparse graph classes.

There are (more recently introduced) structural graph parameters which also generalize the vertex cover number but, in contrast with e.g. the tree-width, these parameters focus on dense graphs. First, up to our knowledge, of these parameters is the *neighborhood diversity* defined by Lampis [17]. We denote the neighborhood diversity of a graph $G = (V, E)$ as $nd(G)$.



■ **Figure 1** A map of the considered parameters. A black arrow stands for a linear upper bound, while a gray arrow stands for an exponential upper bound. That is, if a graph G has $vc(G) \leq k$ then $nd(G) \leq 2^k + k$.

Neighborhood Diversity. We say that two distinct vertices u, v are of the same *neighborhood type* if they share their respective neighborhoods, that is, when $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

► **Definition 5** (Neighborhood diversity [17]). A graph $G = (V, E)$ has *neighborhood diversity* at most w ($nd(G) \leq w$) if there exists a decomposition $\mathcal{D}_{nd} = (C_i)_{i=1}^w$ of $V = C_1 \dot{\cup} \dots \dot{\cup} C_w$ (we call the sets C_i *types*) such that all vertices in a type have the same neighborhood type.

Note that every type induces either a clique or an independent set in G and two types are either joined by a complete bipartite graph or no edge between vertices of the two types is present in G . Thus, we use the notion of a *type graph* – that is a graph T_G representing the graph G and its neighborhood diversity decomposition in the following way. The vertices of type graph T_G are the types C_1, \dots, C_w and two such vertices are joined by an edge if all the vertices of corresponding types are adjacent. We would like to point out that it is possible to compute the neighborhood diversity of a graph in linear time [17].

Twin Cover. If two vertices u, v have the same neighborhood type and $e = \{u, v\}$ is an edge of the graph, we say that e is a *twin edge*.

► **Definition 6** (Twin cover number [14]). A set of vertices $T \subseteq V$ is a *twin cover* of a graph $G = (V, E)$ if for every edge $e \in E$ either $T \cap e \neq \emptyset$ or e is a twin edge. We say that G has *twin cover number* t ($tc(G) = t$) if the size of a minimum twin cover of G is t .

Note that after removing T from a graph G the resulting graph consists of disjoint union of cliques – we call them *twin cliques*. Moreover, for every vertex v in T and a twin clique C holds that v is either adjacent to every vertex in C or to none of them. A *twin cover decomposition* $\mathcal{D}_{tc} = (C_i)_{i=1}^t$ of a graph G is a partition of $V(G)$ such that each C_i is either a vertex of the twin cover or a twin clique.

Note that the twin cover number can be upper-bounded by the vertex cover number. The structure of graphs with bounded twin cover is very similar to the structure of graphs with bounded vertex cover number. Thus, there is a hope that many of known algorithms for graphs with bounded vertex cover number can be easily turned into algorithms for graphs with bounded twin cover number.

Uniform Threshold Function. As it is possible to compute the neighborhood diversity (or the twin cover) decomposition in polynomial time (or FPT-time, respectively), we may assume that the decomposition is given in the input. Given a decomposition \mathcal{D} (\mathcal{D}_{nd} or \mathcal{D}_{tc}) a threshold function $f: V(G) \rightarrow \mathbb{N}$ is *uniform with respect to \mathcal{D}* if $f(u) = f(v)$ for every $u, v \in C$

and every $C \in \mathcal{D}$. Observe that this notion generalizes the previously studied [6] model in which the threshold function is required to satisfy $f(u) = f(v)$ whenever $|N(u)| = |N(v)|$, since this indeed holds if $u, v \in C$ and $C \in \mathcal{D}$. It is not hard to see that the uniform function generalizes both the constant and the majority functions for the twin cover number and the neighborhood diversity; this is true already for the later notion.

Moreover, if $f(v)$ is bounded by a constant c for all $v \in V(G)$, then there exists \mathcal{D}_{nd} with $\|\mathcal{D}_{\text{nd}}\| \leq c \cdot \text{nd}(G)$ such that f is uniform with respect to \mathcal{D}_{nd} . We stress here that this construction is not legal for the twin cover decompositions. **UNIFORM TARGET SET SELECTION** is a variant of **TARGET SET SELECTION**, where the input instance (G, f, b, \mathcal{D}) is restricted in such a way that the function f is uniform with respect to \mathcal{D} .

Modular-width. Both the neighborhood diversity and the twin cover number are generalized by the modular-width. Here we deal with graphs created by an algebraic expression that uses the following four operations:

1. Create an isolated vertex.
2. The *disjoint union* of two graphs, that is from graphs $G = (V, E), H = (W, F)$ create a graph $(V \cup W, E \cup F)$.
3. The *complete join* of two graphs, that is from graphs $G = (V, E), H = (W, F)$ create a graph with vertex set $V \cup W$ and edge set $E \cup F \cup \{\{v, w\} : v \in V, w \in W\}$. Note that the edge set of the resulting graph can be also written as $E \cup F \cup (V \times W)$.
4. The *substitution operation* with respect to a template graph T with vertex set $\{v_1, v_2, \dots, v_k\}$ and graphs G_1, G_2, \dots, G_k created by an algebraic expression; here $G_i = (V_i, E_i)$ for $i = 1, 2, \dots, k$. The substitution operation, denoted by $T(G_1, G_2, \dots, G_k)$, results in the graph on vertex set $V = V_1 \cup V_2 \cup \dots \cup V_k$ and edge set

$$E = E_1 \cup E_2 \cup \dots \cup E_k \cup \bigcup_{\{v_i, v_j\} \in E(T)} \{\{u, v\} : u \in V_i, v \in V_j\}.$$

► **Definition 7** (Modular-width [13]). Let A be an algebraic expression that uses only operations 1–4 above. The *width* of the expression A is the maximum number of operands used by any occurrence of operation 4 in A . The *modular-width* of a graph G , denoted $\text{mw}(G)$, is the least positive integer k such that G can be obtained from such an algebraic expression of width at most k .

An algebraic expression of width $\text{mw}(G)$ can be computed in linear time [22].

Restricted Modular-width. We would like to introduce here a restriction of the modular-width that still generalizes both the neighborhood diversity and the twin cover number. The algebraic expression used to define a graph G may contain the substitution operation at most once and if it contains the substitution operation it has to be the last operation in the expression. However, there is no limitation for the use of operations 1–3.

3 Positive Results

In this section we give proofs of Theorem 1 and 3. In the first part we discuss the crucial property of dense structural parameters – the uniformity of neighborhoods. This, opposed to e.g. the cluster vertex deletion number, allows us to design parameterized algorithms. In this section by a decomposition \mathcal{D} we mean a neighborhood diversity or a twin cover decomposition.

► **Lemma 8.** *Let $G = (V, E)$ be a graph, \mathcal{D} be a decomposition of G , $S \subseteq V$ be a target set, and f be a uniform threshold function with respect to \mathcal{D} . Let $C \in \mathcal{D}$ and $S = S_0, S_1, \dots$ be an activation process arising from S . For each round $i \in \mathbb{N}_0$ one of the following holds either*

1. $S_i \cap C = S_0 \cap C$, or
2. $S_i \cap C = C$.

Moreover, there exist j with $j \in \mathbb{N}_0$ such that for C the first item applies in rounds $0, \dots, j$ and the second in rounds $j + 1, \dots$

Proof. Since f is uniform, it is constant on C . The proof is by induction on the round number i . The statement clearly holds for $i = 0$. Suppose that the statement is valid for all $i' < i$ but not for i , that is, in the i -th round there are two vertices $u, v \in C$ such that $u \in S_i \setminus S_{i-1}$ and $v \notin S_i$. This is impossible, as both u and v have the same neighborhood type and $f(u) = f(v)$. Thus if u gets activated, then v must be activated as well. The “moreover” part follows from the monotonicity of the activation process ($S_i \subseteq S_{i+1}$). ◀

Let $C \in \mathcal{D}$. For a threshold function f which is constant on C we define $f'(C)$ as $f(v)$ for arbitrary vertex v in C . By Lemma 8, we say that C is *activated in a round i* if $S_i \cap C = C$ and $S_j \cap C = S_0 \cap C$ for every $j < i$. We denote $a_i^S(v)$ the number $|S_{i-1} \cap N(v)|$, i.e., the number of active neighbors of v in the round i in the activation process arising from the set S . Thus, a vertex v is activated in the first round i when $a_i^S(v) \geq f(v)$ holds.

3.1 Uniformity and Twin Cover

In this subsection we present an algorithm for UNIFORM TARGET SET SELECTION parameterized by the twin cover number.

Trivial Bounds on the Minimum Target Set. Let $G = (V, E)$ be a graph with twin cover T of size t and let C_1, C_2, \dots, C_q be the twin cliques of G . For a twin clique C by $N(C)$ we denote the common twin cover neighborhood, that is, $N(v) \cap T$ for any $v \in C$. We show that there is a small number of possibilities how the optimal target set can look like. Let $b_C = \max(f'(C) - |N(C)|, 0)$ for a twin clique C .

► **Observation 9.** *If the minimum target set of G has size s , then $B \leq s \leq B + t$ for $B = \sum_{i=1}^q b_{C_i}$.*

Proof. Let S be a target set for G of size s . Suppose there is a twin clique C such that $|S \cap C| = p < b_C$. It means that $b_C > 0$. Let $v \in C \setminus S$. Note that $p < |C|$, thus such a vertex v exists. For the vertex v it holds that $a_i^S(v) < p + |N(C)|$ for every round i of the process. Thus, the vertex v is never activated because $p + |N(C)| < b_C + |N(C)| = f'(C)$ and S is not a target set. On the other hand, if we put b_C vertices from each twin clique C into a set S' , then the set $S' \cup T$ is a target set because every vertex not in S' is activated in the first round. ◀

Structure of the Solution. Let $(G, f, b, \mathcal{D}_{tc})$ be an instance of UNIFORM TARGET SET SELECTION with $tc(G) = t$. By Observation 9, if $b < \sum b_C$, then we automatically reject. On the other hand, if $b \geq t + \sum b_C$, then we automatically accept. Let $w = b - \sum b_C$. Thus, to find a target set of size b we need to select w excess vertices from the twin cliques and the twin cover. We will show there are at most $g(t)$ interesting choices for these w excess vertices for some computable function g and those choices can be efficiently generated. Since we can check if a given set $S \subseteq V(G)$ is a target set in polynomial time, there is an FPT-algorithm for UNIFORM TARGET SET SELECTION.

We start with an easy preprocessing. Let C be a twin clique with $b_C > 0$. We select b_C vertices $V' \subseteq C$ and remove them from the graph G . We also decrease the threshold value by b_C of every vertex which was adjacent to V' (recall that vertices in V' have the same neighborhood type, thus any vertex adjacent to some vertex in V' is adjacent to all vertices in V'). Formally, we get an equivalent instance $(G_1, f_1, b - b_C, \mathcal{D}'_{tc})$, where G_1 is G without vertices V' , \mathcal{D}'_{tc} is \mathcal{D}_{tc} restricted to $V(G_1)$ and

$$f_1(v) = \begin{cases} f(v) & v \notin N_G(V') \\ f(v) - b_C & v \in N_G(V'). \end{cases}$$

It is easy to see that the instances $(G, f, b, \mathcal{D}_{tc})$ and $(G_1, f_1, b - b_C, \mathcal{D}'_{tc})$ are equivalent, because any target set of G needs at least b_C vertices in the twin clique C due to Observation 9. Note that the function f_1 is uniform with respect to \mathcal{D}'_{tc} . We repeat this process for all twin cliques. From now on we suppose that the instance $(G, f, b, \mathcal{D}_{tc})$ is already preprocessed. Thus, for every twin clique C it holds that $b_C = 0$ and $f'(C) \leq N(C) \leq t$.

We say that a twin clique C is of a *type* (Q, r) for $Q \subseteq T, r \leq t$ if $Q = N(C)$ and $f'(C) = r$. Two twin cliques C and D are of the same type if $N(C) = N(D)$ and $f'(C) = f'(D)$. Note that there are at most $(t + 1) \cdot 2^t$ distinct types of the twin cliques.

We start to create a possible target set S of size b . We add w_1 (for some $w_1 \leq w$) vertices from the twin cover T to S (there are at most 2^t such choices). Now we need to select $w_2 = w - w_1$ excess vertices from twin cliques to S .

The number of the twin cliques of one type may be large. Thus, for the twin cliques we need some more clever way than try all possibilities. The intuition is that if we want to select some excess vertices from a clique of a type (Q, r) it is a “better” choice to select the vertices from large cliques of the type (Q, r) . We assign to each type (Q, r) a number $w_{(Q,r)}$ how many excess vertices would be in twin cliques of type (Q, r) . We prove that it suffices to distribute $w_{(Q,r)}$ excess vertices among the $w_{(Q,r)}$ largest twin cliques of the type (Q, r) .

► **Definition 10.** Let C_1, \dots, C_p be all twin cliques of type (Q, r) ordered by the size in a descending order, i.e., for all $i < p$ holds that $|C_i| \geq |C_{i+1}|$. We say that a target set has a *hole* (C_i, C_j) for $i < j$ if $|S \cap C_i| = 0$ and $|S \cap C_j| \geq 1$. A target set is (Q, r) -*leaky* if it has a hole and it is (Q, r) -*compact* otherwise.

Our goal is to prove that if there is a target set S which is (Q, r) -leaky, then there is also a target set R which is (Q, r) -compact and $|R| = |S|$.

► **Lemma 11.** *Suppose there is a target set S for a graph G with a threshold function f and S is (Q, r) -leaky for some twin clique type (Q, r) . Then, there is a target set R such that:*

1. *It holds that $|R| = |S|$.*
2. *The sets R and S differ only at the twin cliques of the type (Q, r) .*
3. *The set R is (Q, r) -compact.*

Proof Sketch. Let set S has a hole (C_i, C_j) for the twin cliques of the type (Q, r) . We create a target set R by removing vertices from C_j and adding the same number of vertices from C_i . Formally, $R = (S \setminus C_j) \cup X$, where $X \subseteq C_i$ and $|X| = |S \cap C_j|$. The verification that R is a target set is technical but rather straightforward. The whole proof is in the full version of the paper. ◀

If we repeat Lemma 11 for every type (Q, r) , we get a target set without any hole. To summarize how to distribute w excess vertices:

1. Pick w_1 vertices from the twin cover T , in total 2^t choices.
2. Distribute $w_2 = w - w_1$ excess vertices among $t \cdot 2^t$ types of twin cliques, in total $(t \cdot 2^t)^t = 2^{\mathcal{O}(t^2)}$ choices.
3. Distribute $w_{(Q,r)}$ excess vertices among the $w_{(Q,r)}$ largest cliques of type (Q, r) , in total t^t choices.

By this we create $2^{\mathcal{O}(t^2)}$ candidates for a target set. For each candidate we test whether it is a target set for G or not. If any candidate is a target set, then we find a target set of size b . If no candidate is a target set, then by argumentation above we know the graph G has no target set of size b . This finishes the proof of Theorem 3.

3.2 Neighborhood diversity

In this section we prove that the UNIFORM TARGET SET SELECTION problem admits an FPT algorithm on graphs of the bounded neighborhood diversity. We again use Lemma 8. Note that in each round of the activation process at least one type has to be activated. This implies that there are at most $\text{nd}(G)$ rounds of the activation process. We use this fact to model the whole activation process as an integer linear program which is then solved using Lenstra’s celebrated result:

► **Proposition 12** ([18, 12]). *Let p be the number of integral variables in a mixed integer linear program and let L be the number of bits needed to encode the program. Then it is possible to find an optimal solution in time $\mathcal{O}(p^{2.5p} \text{poly}(L))$ and a space polynomial in L .*

There has to be an order in which the types are activated in order to activate whole graph. Since there are $t = \text{nd}(G)$ types, we can try all such orderings. Let us fix an order \prec on types. To construct the ILP we further need to know which types are fully activated at the beginning. Denote by c_0 the number of such types. Once the order \prec is fixed the set of fully activated types at the beginning is determined by c_0 . Since c_0 can attain values $0, \dots, t$ we can try all $t + 1$ possibilities. Now, with both \prec and c_0 fixed, denote the set of the types activated in the beginning by \mathcal{T}_0 .

Observe further that, as the vertices in a type share all neighbors, the only thing that matters is the number of activated vertices in each type and not the actual vertices activated. Thus, we have variables x_C which corresponds to the number of vertices in type C selected into a target set S .

Let C be a type and n_C be the number of vertices in C . Since we know when C is activated, we know how many active vertices are in C in each round. There are x_C vertices before the activation of C and n_C after the activation. To formulate the integer linear program we denote the set of types by \mathcal{T} and we write $D \in N(C)$ if the two corresponding vertices in the type graph T_G are connected by an edge.

ILP Formulation.

$$\begin{aligned}
 & \text{minimize} && \sum_{C \in \mathcal{T}} x_C \\
 & \text{subject to} && f'(C) \leq \sum_{D \prec C, D \in N(C)} n_D + \sum_{D \succ C, D \in N(C)} x_D + [C \text{ is a clique}]x_C \quad \forall C \in \mathcal{T} \setminus \mathcal{T}_0 \\
 & \text{where} && 0 \leq x_C < n_C \quad \forall C \in \mathcal{T} \setminus \mathcal{T}_0 \\
 & && x_C = n_C \quad \forall C \in \mathcal{T}_0
 \end{aligned}$$

As there are at most $t!$ orders of the set \mathcal{T} and $t + 1$ choices of c_0 this implies that the UNIFORM TARGET SET SELECTION problem can be solved in time $(t + 1)t!t^{\mathcal{O}(t)} \text{poly}(n) = t^{\mathcal{O}(t)} \text{poly}(n)$. Thus, we have proven Theorem 1.

4 Hardness Reductions

In this section we prove that TARGET SET SELECTION is $W[1]$ -hard on graphs of the bounded neighborhood diversity for a general threshold function. We use an FPT-reduction from k -MULTICOLORED CLIQUE.

k -MULTICOLORED CLIQUE

Parameter: k

Input: A k -partite graph $G = (V_1 \cup \dots \cup V_k, E)$, where V_c is an independent set for every $c \in [k]$ and they are pairwise disjoint.

Task: Find a clique of the size k .

Let G be an input of k -MULTICOLORED CLIQUE. We refer to a set V_c as to a *color class* of G and to a set E_{cd} as to edges between color classes V_c and V_d . The problem is $W[1]$ -hard [9] even if every color class V_c has the same size and the number of edges between every V_c and V_d is the same. For an easier notation, we denote the size of an arbitrary color class V_c by $n + 1$ and the size of an arbitrary set E_{cd} by $m + 1$. We describe a reduction from the graph G to an instance of TARGET SET SELECTION $(G', f: V \rightarrow \mathbb{N}, b)$ where $\text{nd}(G')$ is $\mathcal{O}(k^2)$. The reduction runs in time $\text{poly}(|G|)$. The graph G has a clique of size k if and only if the graph G' has a target set of size b .

In the k -MULTICOLORED CLIQUE problem we need to select exactly one vertex from each color class V_c and exactly one edge from each set E_{cd} . Moreover, we have to make certain that if $\{u, v\} \in E_{cd}$ is a selected edge, then $u \in V_c$ and $v \in V_d$ are selected vertices.

An Overview of Proof of Theorem 2. We present a way of encoding a vertex v in a color class V_c of the graph G by two numbers v -pos and v -neg with v -pos + v -neg = n . We proceed with encoding of edges similarly, however, edges are encoded by multiples of sufficiently large number q . This we do in such a way that sum of the encoding of a vertex and an incident edge is unique. We create three types of gadgets: for selection vertices, for selection edges, and gadgets which check that the selected vertices are incident to the selected edges. Proofs of lemmas and theorems in this section are quite technical and they are presented in the full version of this paper.

4.1 Proof of Theorem 2

In order to present the hardness reduction we have to introduce some gadgets. We denote the name of a gadget by capital letters and we write parameters of the gadget into parentheses (e.g. $L(s)$). When speaking about concrete instance of a gadget kind $L(s)$ we add a subscript, i.e., $L_c(s)$. We omit parameters of the gadget if they are clear from the context.

Selection Gadget. First, we describe gadgets of the graph G' for selecting vertices and edges of the graph G . For an overview of the reduction see Figure 2. The gadget $L(s)$ is formed by two types L -neg and L -pos of equal size s (the number s will be determined later); we refer to these two types as the *selection part*. For a vertex v in the selection part we set the value $f(v)$ of the threshold to the degree of v . It means that if some vertex v from the selection part is not selected into the target set, then all neighbors of v have to be active before the vertex v can be activated by the activation process. The selection gadget L is connected to the rest of the graph using only vertices from the selection part.

18:10 Target Set Selection in Dense Graph Classes

The last part of the gadget L is formed by type L -guard of $s + 1$ vertices connected to both types in the selection part. For each vertex v in L -guard type we set $f(v) = s$.

Numeration of Vertices and Edges. Now, we describe how we use the selection gadget. Let $V_c = \{v_0, \dots, v_n\}$. For every color class V_c we create a selection gadget $L_c = L(n)$. We select a vertex $v_i \in V_c$ to the multicolor clique if i vertices in the L_c -pos type and $n - i$ vertices in the L_c -neg type of the gadget L_c are selected into the target set.

The selection of edges is similar, however, a bit more complicated. Let $q \in \mathbb{N}$ and $E_{cd} = \{e_0, \dots, e_m\}$. For every set E_{cd} we create a selection gadget L_{cd} of kind $L(qm)$. We select an edge $e_j \in E_{cd}$ to the multicolor clique if qj vertices in the L_{cd} -pos type of the gadget L_{cd} are selected into the target set (and $q(m - j)$ vertices in the L_{cd} -neg are selected into the target set). Suppose s vertices in the L_{cd} -pos type are selected into the target set. If s is not divisible by q , then it is an invalid selection. We introduce a new gadget which controls that s has to be divisible by q .

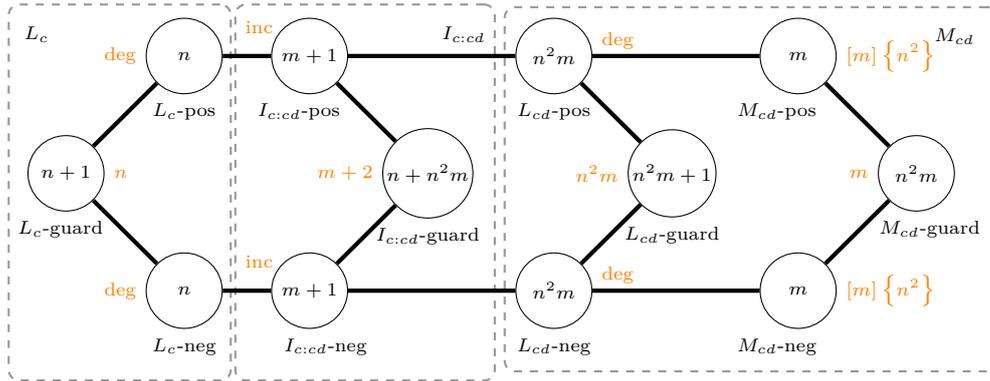
Multiple Gadget. A multiple gadget $M(q, s)$ consists of a selection gadget $L(qs)$ and 3 other types: M -pos, M -neg of s vertices and M -guard of qs vertices. The type M -pos is connected to the type L -pos and the type M -neg is connected to the type L -neg. The type M -guard is connected to the types M -pos and M -neg. Still, the rest of graph G' is connected only to types L -pos and L -neg. Let $\{u_1, \dots, u_s\}$ and $\{w_1, \dots, w_s\}$ be vertices in M -pos type and M -neg type, respectively. We set thresholds $f(u_i) = f(w_i) = qi$. For each vertex v in M -guard we set $f(v) = s$.

Incidence Gadget. So far we described how we encode in graph G' selecting vertices and edges to multicolor clique. It remains to describe how we encode the correct selection, i.e., if $v \in V_c$ and $e \in E_{cd}$ are selected vertex and edge to multicolor clique, then $v \in e$. We create $L_c(n)$ selection gadget for a color class V_c . We set the number q to n^2 and create a multiple gadget M_{cd} of kind $M(n^2, m)$ (with selection gadget L_{cd}) for a set E_{cd} . We join gadgets L_c and M_{cd} through an incidence gadget $I_{c:cd}$. The incidence gadget $I_{c:cd}$ has three types $I_{c:cd}$ -pos and $I_{c:cd}$ -neg of $m + 1$ vertices and $I_{c:cd}$ -guard of $n + n^2m$ vertices. We connect the $I_{c:cd}$ -guard type to the types $I_{c:cd}$ -pos and $I_{c:cd}$ -neg. Furthermore, we connect the type $I_{c:cd}$ -pos to the types L_c -pos and L_{cd} -pos. Similarly, we connect the type $I_{c:cd}$ -neg to the types L_c -neg and L_{cd} -neg.

We set thresholds of all vertices in the $I_{c:cd}$ -guard type to $m + 2$. Recall there are $m + 1$ edges in the set E_{cd} . Thus, we can associate edges in E_{cd} with vertices in $I_{c:cd}$ -pos ($I_{c:cd}$ -neg respectively) one-to-one. I.e., $V(I_{c:cd}$ -pos) = $\{u_\ell : e_\ell \in E_{cd}\}$ and $V(I_{c:cd}$ -neg) = $\{w_\ell : e_\ell \in E_{cd}\}$. Let $v_i \in V_c, e_j \in E_{cd}$ and $v_i \in e_j$. Recall that selecting v_i and e_j into a multicolor clique is encoded as selecting i vertices in L_c -pos type and n^2j vertices in L_{cd} -pos type into a target set. We set threshold of u_j to $i + n^2j$ and threshold of w_j to the “opposite” value $n - i + n^2(m - j)$.

Since we set the coefficient q to n^2 , for each edge $e_j \in E_{cd}$ and each vertex $v_i \in V_c$ the sum $i + n^2j$ is unique. Thus, every vertex in $I_{c:cd}$ -pos ($I_{c:cd}$ -neg) has a unique threshold. We will use this number to check the incidence.

Reduction Correctness. We described how from the graph G with k color classes (instance of k -MULTICOLORED CLIQUE) we create the graph G' with the threshold function f (input for TARGET SET SELECTION). For every color class V_c we create a selection gadget L_c . For



■ **Figure 2** An overview of the reduction. The number inside a type is the number of vertices of the type. The threshold of vertices in a type is displayed next to the type in orange (light-gray).

every edge set E_{cd} we create a multiple gadget M_{cd} . We join the gadgets L_c and M_{cd} by an incidence gadget $I_{c:cd}$ (gadgets L_d and M_{cd} are joined by a gadget $I_{d:cd}$). It is easy to see the following observations by constructions of G' .

- ▶ **Observation 13.** *The graph G' has polynomial size in the size of the graph G .*
- ▶ **Observation 14.** *Neighborhood diversity of the graph G' is $\mathcal{O}(k^2)$.*

To finish the construction of an instance of TARGET SET SELECTION, we set the budget b to $kn + \binom{k}{2}n^2m$. The main idea of proofs of the following theorems is that we select a vertex $v_i \in V_c$ (or an edge $e_j \in E_{cd}$) into a clique if and only if we select i vertices from the L_c -pos type (or n^2j vertices from the L_{cd} -pos type). Theorem 2 is a corollary of Observation 13, 14 and the following theorem.

- ▶ **Theorem 15.** *The graph G contains a clique of size k if and only if the graph G' with the threshold function f contains a target set of size b .*

4.2 Overview of Proof of Theorem 4

In fact this can be seen as a clever twist of the ideas contained in the proof of Theorem 2. There are some nodes of the neighborhood diversity decomposition already operating in the majority mode – e.g. guard vertices – these we keep untouched. For vertices with threshold set to their degree one has to “double” the number of vertices in the neighborhood. Finally, one has to deal with types having different thresholds for each of its vertices, which is quite technical.

5 Conclusions

We have generalized ideas of previous works [3, 19] for the TARGET SET SELECTION problem. The presented results give a new idea how to encode selecting vertices and edges in the k -MULTICOLORED CLIQUE problem for showing $W[1]$ -hardness. In particular, only few problems are known to be $W[1]$ -hard when parameterized by neighborhood diversity – which is the case for the TARGET SET SELECTION problem.

Thus, we would like to address an open problem regarding structural parameterizations of the TARGET SET SELECTION problem. Determine parameterized complexity of the TARGET SET SELECTION problem parameterized by twin cover number. Furthermore, we are not

aware of other positive results concerning the number of different thresholds instead of the threshold upper-bound.

We would like to point out that in our proofs of $W[1]$ -hardness the activation process terminates after constant number of rounds (independent of the parameter value and the size of the input graph). This is true also for all reductions given by Chopin et al. [6].

References

- 1 Eyal Ackerman, Oren Ben-Zwi, and Guy Wolfowitz. Combinatorial model and bounds for target set selection. *Theoretical Computer Science*, 411(44):4017–4022, 2010. doi:10.1016/j.tcs.2010.08.021.
- 2 József Balogh, Béla Bollobás, and Robert Morris. Bootstrap Percolation in High Dimensions. *Combinatorics, Probability & Computing*, 19(5-6):643–692, 2010. doi:10.1017/S0963548310000271.
- 3 Oren Ben-Zwi, Danny Hermelin, Daniel Lokshantov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011. Parameterized Complexity of Discrete Optimization.
- 4 Ning Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.
- 5 Chun-Ying Chiang, Liang-Hao Huang, Bo-Jr Li, Jiaojiao Wu, and Hong-Gwa Yeh. Some results on the target set selection problem. *Journal of Combinatorial Optimization*, 25(4):702–715, 2013. doi:10.1007/s10878-012-9518-3.
- 6 Morgan Chopin, André Nichterlein, Rolf Niedermeier, and Mathias Weller. Constant Thresholds Can Make Target Set Selection Tractable. *Theory Comput. Syst.*, 55(1):61–83, 2014. doi:10.1007/s00224-013-9499-3.
- 7 Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanič, Joseph Peters, and Ugo Vaccaro. Spread of influence in weighted networks under time and budget constraints. *Theoretical Computer Science*, 586:40–58, 2015.
- 8 Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanič, and Ugo Vaccaro. Latency-bounded target set selection in social networks. *Theoretical Computer Science*, 535:1–15, 2014.
- 9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 10 Pedro Domingos and Matt Richardson. Mining the network value of customers. In *ACM SIGKDD*, pages 57–66. ACM, 2001.
- 11 Paul A. Dreyer Jr. and Fred S. Roberts. Irreversible k -threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615–1627, 2009. doi:10.1016/j.dam.2008.09.012.
- 12 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. doi:10.1007/BF02579200.
- 13 Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized Algorithms for Modular-Width. In *IPEC 2013*, pages 163–176, 2013. doi:10.1007/978-3-319-03898-8_15.
- 14 Robert Ganian. Twin-Cover: Beyond Vertex Cover in Parameterized Algorithmics. In *IPEC 2011*, pages 259–271, 2011. doi:10.1007/978-3-642-28050-4_21.
- 15 Tim A. Hartmann. Target Set Selection Parameterized by Clique-Width and Maximum Threshold. In *SOFSEM 2018*, pages 137–149. Springer International Publishing, 2018.

- 16 David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD*, pages 137–146. ACM, 2003.
- 17 Michael Lampis. Algorithmic Meta-theorems for Restrictions of Treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.
- 18 Hendrik W. Lenstra, Jr. Integer Programming with a Fixed Number of Variables. *Mathematics of Operations Research*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.
- 19 André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. On tractable cases of Target Set Selection. *Social Netw. Analys. Mining*, 3(2):233–256, 2013. doi:10.1007/s13278-012-0067-7.
- 20 David Peleg. Local Majorities, Coalitions and Monopolies in Graphs: A Review. *Theor. Comput. Sci.*, 282(2):231–257, 2002. doi:10.1016/S0304-3975(01)00055-X.
- 21 Matthew Richardson and Pedro Domingos. Mining Knowledge-sharing Sites for Viral Marketing. In *ACM SIGKDD*, KDD '02, pages 61–70, New York, NY, USA, 2002. ACM. doi:10.1145/775047.775057.
- 22 Marc Tedder, Dereck G. Corneil, Michel Habib, and Christophe Paul. Simpler Linear-Time Modular Decomposition Via Recursive Factorizing Permutations. In *ICALP 2008*, pages 634–645, 2008. doi:10.1007/978-3-540-70575-8_52.