Target Set Selection in Dense Graph Classes

Pavel Dvořák**, Dušan Knop***, and Tomáš Toufar

Computer Science Institute, Charles University, Prague, Czech Republic
koblich@iuuk.mff.cuni.cz
Department of Applied Mathematics, Charles University, Prague, Czech Republic
knop@kam.mff.cuni.cz
Computer Science Institute, Charles University, Prague, Czech Republic
toufi@iuuk.mff.cuni.cz

Abstract. In this paper we study the Target Set Selection problem, a fundamental problem in computational social choice, from a parameterized complexity perspective. Here for a given graph and a threshold for each vertex the task is to find a set of active vertices that activates whole graph. A vertex becomes active if the number of activated vertices in its neighborhood is at least its threshold.

We give two parameterized algorithms for a special case where each vertex has threshold set to half of its neighbors (the so called Majority Target Set Selection problem) for parameterizations by neighborhood diversity and twin cover number of the input graph. From the opposite side we give hardness proof for the Majority Target Set Selection problem when parameterized by (restriction of) the modular-width – a natural generalization of both previous structural parameters. Finally, we give hardness proof for the Target Set Selection problem parameterized by the neighborhood diversity when there is no restriction on the thresholds.

1 Introduction

We study the Target Set Selection problem, introduced by Kempe et al. [12], from the area of computational social choice from a parameterized complexity perspective. The three most important areas of research in the computational social choice are theoretical computer science, logic, and artificial intelligence.

Target set selection. Let $G = (V, E)$ be a graph, $S \subseteq V$, and threshold function $f: V \to \mathbb{N}$. The activation process arising from the set $S_0 = S$ is an iterative process with resulting sets $S_0, S_1, \ldots$ such that for $i \geq 0$

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

---

* Research partly supported by the project SVV–2016–260332.
** 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.
*** Research partly supported by the CE-ITI grant project P202/12/G061 of GA ČR, by GAUK project 1784214.
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 \( i > 0 \). Let \( i \) be smallest integer for which \( S_i = S_{i+1} \) holds we say that the activation process terminates in round \( i \). We say that the set \( S \) is a target set and activation process \( S = S_0, \ldots, S_n \) is successful if \( S_n = V \).

**Target Set Selection**

*Input:* graph \( G = (V, E) \), \( f : V \rightarrow \mathbb{N} \) and positive integer \( k \in \mathbb{N} \)

*Task:* find a target set \( S \subseteq V \) of size at most \( k \)

The problem interpretation and computational complexity clearly may vary depending on the input function \( f \). There are three important setting studied – namely constant, majority, and general function. If the threshold function \( f \) is a 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**.

**Computational Social Choice**

Computational social choice is an interdisciplinary field of study at the interface of social choice theory and computer science, promoting an exchange of ideas in both directions. Computational social choice is concerned with importing concepts from social choice theory into computing. For instance, social welfare orderings originally developed to analyse the quality of resource allocations in human society are equally well applicable to problems in multiagent systems or network design.

**Distance to Triviality.** There are many natural parameters assumed nowadays in the parameterized complexity studies – among these the size of the solution set and the structural parameters play the most significant role. Sometimes another very important parameter – distance to triviality – is assumed. There are two major examples of this parameter use either the parameter of value \( k \) expresses that after removal of \( k \) vertices the input graph is turned in a graph belonging to a class of graphs on which the assumed problem becomes trivial (polynomial time solvable) or it may be viewed as the distance from guarantee, as for example the guarantee given by rounding a relaxation of the integer linear program [14]. In this work we use the structural parameters suitable for dense graphs, however Chopin et al. [5] already observed that the **Target Set Selection** problem can be trivially solved on cliques and thus our structural parameters may be viewed as a distance to triviality in this context.

**Previous Results.** Target set solution received an attention of researchers in theoretical computer science in the past years [1,2,3,4,5,12,15]. A general lower 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 [15]. An \( t^{O(w)} \) \( \text{poly}(n) \) algorithm (an XP algorithm) where \( w \) is the treewidth of the input graph and \( t \) is an upper-bound on the threshold function [2], that is \( f(v) \leq t \) for every vertex \( v \). This is essentially optimal, as the **Target Set Selection** problem is \( W[1] \)-hard for majority

\[ \text{https://www.illc.uva.nl/COMSOC/} \]
and general functions \[5\]. The Target Set Selection problem is solvable in
linear time on trees \[3\] and more general on block-cactus graphs \[4\]. The problem
is also hard to approximate \[3\] within a polylogarithmic factor. For more and
not recent results we refer the reader to a survey by Peleg \[16\].

**Our Results.** In this work we generalize the results obtained by Chopin et
al. \[5\] who studied the Target Set Selection on various graph classes. They
essentially proved that in sparse graph classes (such as 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. However, this is not true in
general for dense graph classes. This we demonstrate for the parameterization
by neighborhood diversity:

**Theorem 1.** There is an FPT algorithm for the Majority Target Set Se-
lection 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 Majority Target Set Selection problem is not resolved
for parametrization by the cluster vertex deletion number (the number of vertices
whose removal from the graph results in a collection of disjoint cliques). We have
a positive result in this direction that also assumes that each vertex we remove is
completely adjacent to the whole clique or completely nonadjacent. This result
also suggest 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 Majority Target Set Se-
lection problem parameterized by the size of the twin cover.

On the contrary, the parameterized complexity of the two problems is again
the same in graphs with bounded clique-width. 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 a formal definition of several graph parameters used in this work. For a
better acquaint with these parameters, we provide a map of assumed parameters
in Figure \[1\].
Definition 1 (Vertex cover). For a graph $G = (V, E)$ the set $U \subset V$ 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 $\text{vc}(G)$, is the least integer $k$ for which there exists a vertex cover of size $k$.

We say that the vertex cover number is very restrictive graph parameter, because for a fixed positive integer $k$ the class of graphs with vertex cover number bounded by $k$ does not contain large spectra of graphs.

As the vertex cover number is (usually) too restrictive, many authors focused on defining other 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 bounded tree-width (respectively path-width) are contained in the so called sparse graph classes.

There are (more recent) structural graph parameters which also generalize the vertex cover number but in contrary to the tree-width these parameters focus on dense graphs. First, up to our knowledge, of these parameters is the neighborhood diversity defined by Lampis [13]. We denote the neighborhood diversity of a graph $G = (V, E)$ as $\text{nd}(G)$.

**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 2 (Neighborhood diversity [13]). A graph $G = (V, E)$ has neighborhood diversity at most $w$ ($\text{nd}(G) \leq w$) if there exists a partition of $V$ into at most $w$ sets (we call these sets 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 neighborhood types of the graph $G$ and two such vertices are joined by an edge if all the vertices of corresponding types are joined by an edge. We would like to point out that it is possible to compute the neighborhood diversity of a graph in linear time [13].
**Twin cover.** More recently, Ganian [9] defined the twin cover number. We begin with an auxiliary definition. 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 3 (Twin cover number [9]).** 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

1. $T \cap e \neq \emptyset$, or
2. $e$ is a twin edge.

We say that $G$ has twin cover number $k$ ($\text{tc}(G) = k$) if the size of a minimum twin cover of $G$ is $k$.

Note that after removing $T$ from a graph $G$ the resulting graph consists of disjoint union of cliques. We denote these cliques as twin cliques.

Note that the twin cover can be upper-bounded by the vertex cover number. As the structure of graphs with bounded twin cover is very similar to the structure of graphs with bounded vertex cover number 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.

**Modular-width.** Both neighborhood diversity and twin cover number are generalized by a modular-width, defined by Gajarský et al. [8]. Here we deal with graphs created by an algebraic expression that uses four following 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 $\{v_1, v_2, \ldots, v_k\}$ and graphs $G_1, G_2, \ldots, G_k$ created by algebraic expression. The substitution operation, denoted by $T(G_1, G_2, \ldots, G_k)$, results in the graph on vertex set $V = V_1 \cup V_2 \cup \cdots \cup V_k$ and edge set

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

where $G_i = (V_i, E_i)$ for all $i = 1, 2, \ldots, k$.

**Definition 4 (Modular-width [8]).** Let $A$ be an algebraic expression that uses only operations 1–4. The width of 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 as $\text{mw}(G)$, is the least positive integer $k$ such that $G$ can be obtained from such an algebraic expression of width at most $k$. 5
When a graph $H$ is constructed by the fourth operation, that is $G = T(G_1, G_2, \ldots, G_k)$, we call the graph $T$ the **template graph**. An algebraic expression of width $\text{mw}(G)$ can be computed in linear time [17].

**Restricted Modular-width.** We would like to introduce here a restriction of the modular width that still generalizes both neighborhood diversity and twin cover number. The algebraic expression used to define graph $G$ can 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.

It is easy to see that this generalizes neighborhood diversity as it is possible to build a complete graph or independent set of arbitrary size using operations 1 and 3 only. A graph $G$ with $\text{nd}(G) \leq k$ can be constructed from a type graph $T_G$ by replacing each vertex in $V(T_G)$ by an independent set or a clique. A graph $G$ with $\text{rmw}(G) \leq k$ can be constructed from a type graph $T_G$ by replacing each vertex in $V(T_G)$ by arbitrary cograph – a graph $H$ is a cograph if $H$ can be constructed by operations 1–3. Since cliques and independent sets are cographs, restricted modular-width is generalization of neighborhood diversity.

It is not hard to argue that this parameter generalizes twin cover number as well. To see this divide the twin cliques according to their neighborhood in the set $T$. Now observe that it is possible to build disjoint union of cliques using operation 2.

One may ask, whether requiring the template operation to be used just once leads to the same class of graph as if we require it to be the last operation. However, this is not the case as is shown in the following lemma. Thus, we obtain sort of hierarchy leading to modular-width.

**Lemma 1.** Requiring the substitution operation to be used as the last operation is more restrictive than requiring it to be used just once.

**Proof.** To show this we build a family of graphs $G_n$ on $n = 4, 5, \ldots$ vertices which admits a decomposition in which only one substitution operation of width 4 is used. We begin by setting $G_4 = P_4$. Note that $P_4$ is not a cograph and thus it has to be constructed using the substitution operation. We define for $n > 2$ graph $G_{2n+1}$ as the graph $G_{2n}$ plus an apex vertex and $G_{2n+2}$ as the graph $G_{2n+1}$ with an isolated vertex added.

Now observe that if we require the substitution operation to be the last operation of the decomposition of $G_n$, then its width is $n$. While if we allow to build $P_4$ using substitution operation of size 4, then what remains can be build only using the operations 1–3. \qed

### 3 Positive Results

In this section we give proofs of Theorem 3 and 1. In the first part we discuss the crucial property of these structural parameters – uniformity of neighborhood. This, opposed to e.g. cluster vertex deletion number, allows us to design
parameterized algorithm. We study parameters twin-cover and neighborhood diversity.

Lemma 2. Let $G = (V, E)$ be a graph, $S \subseteq V$. Now let $C$ be an independent set or a clique such that every two vertices of $C$ have the same neighborhood type. Suppose the threshold function $f$ is constant on $C$. Let $S_0 = S, S_1, \ldots, S_r$ be the activation process arising from $S$. For each round $i \in \{0, 1, \ldots, r\}$ one of the following holds:

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

Moreover, there exist $j$ with $0 \leq j \leq r$ such that for $C$ the first item applies in rounds $0, \ldots, j$ and the second in rounds $j + 1, \ldots, r$.

Proof. The proof is by induction on the round number $i$. The statement clearly holds for $i = 0$. Suppose that lemma is valid for all $i' < i$ but not for $i$—this means that in the $i$-th round there are two vertices $u, v \in C$ such that $u \in S_i \setminus S_{i-1}$ but $v \notin S_i$. This is impossible as both $u$ and $v$ have the same threshold and the same neighborhood type. Thus if $u$ gets activated, then $v$ must be activated as well. The moreover part also follows easily. \(\square\)

Let $C$ be a twin clique or type of neighborhood diversity. Note that $C$ fulfill the neighborhood type condition in Lemma 2. For a threshold function $f$ which is constant on $C$ we define $f'(C)$ as $f(v)$ for arbitrary vertex in $C$. By Lemma 2 we say that $C$ is activated in round $i$ if $C \cap S_i = C$ and $C \cap S_j = S$ for every $j < i$. We denote the round when a twin clique $C$ is activated by $A_S(C)$. Further, we denote $a^S_i(v)$ the number $|S_i \cap N(v)|$, i.e., the number of active neighbors of $v$ in the round $i$ in the activation process arising from the target set $S$. Thus, a vertex $v$ is activated in the first round $i$ when holds $a_i(v) \geq f(v)$.

### 3.1 Majority and Twin Cover

In this subsection we assume the threshold function $f$ is the majority function. Note that $f$ is constant on each twin clique, thus we can use Lemma 2 for this setting.

**Trivial Bounds on the Minimum Target Set.** Let $G = (V, E)$ be a graph with twin cover $T$ of size $t$ and let $C_1, C_2, \ldots, C_q$ 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 $v \in V(C)$. We show that there are only small number of possibilities how the optimal target set can look like. Let $k_C = \max(f'(C) - |N(C)|, 0)$ for a twin clique $C$.

**Observation 5** Suppose the minimum target set of $G$ has size $s$. For $k' = \sum_{i=1}^q k_{C_i}$ holds that $k' \leq s \leq k' + t$.
Proof. Suppose there is a twin clique $C$ such that $|S \cap C| = p < k_C$. It means that $k_C > 0$. Let $v \in V(C) \setminus S$. Note that $k_C < |V(C)|$, thus the vertex $v$ exists. For vertex $v$ holds that $a^*_v(v) < p + |N(C)|$ for every round $i$ of the process. Thus, the vertex $v$ is never activated because $p + |N(C)| < k_C + |N(C)| = f(C)$.

On the other hand, if we put $k_C$ vertices from each twin clique $C$ into a set $S'$, then the set $S' \cup T$ is a target set. $\square$

**Structure of the Solution.** Let $(G, f, k)$ be an instance of Majority Target Set Selection with $tc(G) = t$. By Observation 5 if $k < \sum k_C$, then we automatically reject. On the other hand, if $k \geq t + \sum k_C$, then we automatically accept. Let $w = k - \sum k_C$. Thus, to found a target set of size $k$ we need to select $w$ excess vertices from twin cliques and twin cover. We will show there are at most $g(t)$ interesting choices how to select these $w$ excess vertices for some computable function $g$. Since we can check if a given set $S \subseteq V(G)$ is a target set in polynomial time, we will have an FPT-algorithm for Majority Target Set Selection.

We start to create a possible target set $S$ of size $k$. First, we put $k_C$ vertices from each twin clique $C$ into $S$. We add $w_1$ (for some $w_1 \leq w$) vertices from twin cover to $S$ (at most $2^t$ choices). Now we need to select $w_2 = w-w_1$ excess vertices from twin cliques to $S$. However, the number of twin cliques is big. Thus, for twin cliques we need some more clever way than try all possibilities.

We say that a twin clique $C$ is of type $Q \subseteq T$ if $Q = N(C)$. Two twin cliques $C$ and $D$ are of the same type if $N(C) = N(D)$. Note that there are at most $2^t$ distinct types of twin cliques. Thus, we assign each type $Q$ a number $w_Q$ how many excess vertices would be in twin cliques of type $Q$.

**Observation 6** Let $S = S_0, \ldots, S_r$ be an activation process in the graph $G$. Let $i > 0$ be a round when the first twin clique of type $Q$ is activated. Then, the first $i - 1$ rounds of the activation process does not depend on how $w_Q$ excess vertices are distributed among the twin cliques of type $Q$.

Proof. Let $Q$ be twin cliques of type $Q$. In the first $i - 1$ rounds of the activation process no clique in $Q$ is activated. Let $v \in Q$ (recall $Q \subseteq T$). The vertex $v$ has $p = w_Q + \sum_{C \in Q} k_C$ active neighbors among the vertices in $V(Q)$ during the first $i - 1$ round. Formally, for every $j < i$ holds that $|V(Q) \cap N(v) \cap S_j| = p$. Therefore, if the vertex $v$ is activated or not during the first $i - 1$ round does not depend on how the excess vertices are distributed among the clique in $Q$. $\square$

By Observation 5 we know we can distribute $w_Q$ excess vertices to cliques of type $Q$ arbitrarily and the beginning of the activation process is still the same. Thus, it make sense to try to activate the clique from largest to smallest.

We say a twin clique $C$ is small if $k_C = 0$ (i.e., $f(C) \leq |N(C)|$), otherwise the clique $C$ is big. There is a difference in using excess vertices in big cliques and small cliques. As we proof in the next lemma we can suppose the big cliques of the same type are activated from the largest one to the smallest one. This is not true for small cliques.
**Lemma 3.** Let graph $G$ be a graph with target set $S$. Let $C$ and $D$ be big twin cliques of the same type with $|V(C)| \geq |V(D)|$, the clique $C$ has no excess vertices and $D$ is activated before $C$, i.e., $A_S(D) < A_S(C)$. Then, there exists a target set $S'$ such that $|S'| = |S|$ and $A_{S'}(C) = A_{S'}(D)$.

**Proof.** The twin cliques $C$ and $D$ have the same neighborhood $Q$. Let $e_D = |V(D) \cap S| - k_C$ be a number of excess vertices of target set $S$ in the twin clique $D$. We construct the target set $S'$ by moving the excess vertices from the clique $D$ to clique $C$. I.e., $|S' \cap V(C)| = k_C + e_D$ and $|S' \cap V(D)| = k_D$. Let $i = A_S(D)$.

We analyze the difference between activation processes $P = (S = S_0, \ldots, S_p)$ and $P' = (S' = S'_0, \ldots, S'_p)$. Till the round $i$ the process $P$ runs exactly in the same way as the process $P'$ because for each vertex $v \in Q$ and $j < i$ holds that $a^S_j(v) = a^{S'}_j(v)$. Let $Q' = Q \cap S_{i-1} = Q \cap S'_{i-1}$ be active vertices in $Q$ in the round $i$ of both process. In the round $i$ of the process $P$ the clique $D$ is activated. Thus, there are at least $k_D + |Q|$ active vertices in $Q \cup V(D)$ in the round $i$ of the process $P$, i.e.,

$$|S_i \cap (V(D) \cup Q)| = k_D + e_D + |Q'| \geq k_D + |Q|.$$  

In the process $P'$ there are $k_C + e_D + |Q'| \geq k_C + |Q|$ active vertices in $Q \cup V(D)$ in the round $i$. Since $k_C > 0$ and $k_C = f'(C) - |Q|$, the clique $C$ is activated in the round $i$ of process $P'$.

In the round $i$ of the process $P$ the clique $D$ is activated and the process is successful. In the round $i$ of the process $P'$ the bigger clique $C$ of the same type is activated instead of the clique $D$. Therefore, the process $P'$ is successful as well. \hfill $\square$

We continue with distribution of $w_Q$ excess vertices to twin cliques. We divide $w_Q$ to two numbers $w_Q^a$ and $w_Q^b$. We put $w_Q^a$ excess vertices to small cliques of type $Q$ and $w_Q^b$ excess vertices to big cliques of type $Q$. By Lemma 3 we know we can put $w_Q^b$ excess vertices only to $w_Q^b$ largest twin cliques of type $Q$. Since $w_Q^b \leq t$, there is at most $t^t$ choices how to distribute $w_Q^b$ excess vertices among big cliques of type $Q$.

For small clique $C$ holds that $k_C = 0$, thus $f'(C) \leq |N(C)|$. Since $f'$ is a majority function, $|V(C)| \leq t + 1$. Thus, there are $t + 1$ possible sizes for small cliques. Overall, no more than $w_Q^a$ small cliques of each size can have excess vertices. Thus, there are at most $(w_Q^a)^{w_Q^a(t+1)} \leq t^{O(t^2)}$ choices how to distribute excess vertices into small cliques of specific type. To summarize how to distribute $w$ excess vertices:

1. Pick $w_1$ vertices from twin cover $T$, in total $2^t$ choices.
2. Distribute $w_2 = w - w_1$ excess vertices among $2^t$ types of twin cliques, in total $t \cdot 2^t$ choices.
3. Distribute $w_Q^b$ excess vertices among $w_Q^b$ largest big cliques of type $Q$, in total $t^t$ choices.
4. Distribute $w_Q^a$ excess vertices among small cliques of type $Q$, in total $t^{O(t^2)}$ choices.
Thus, we create $t^O(t^2)$ sets $S$. For each $S \in S$ we decide whether it is a target set or not. If any set $S \in S$ is a target set, then we find a target set of size $k$. If no set in $S$ is a target set, then by argumentation above we know the graph $G$ has no target set of size $k$. This finishes the proof of Theorem 3.

### 3.2 Neighborhood diversity

In this section we prove that the Target Set Selection problem admits an FPT algorithm on graphs of bounded neighborhood diversity whenever the threshold function $f$ is constant on each type. Thus, Lemma 2 holds for this settings. Note that, in each round of the activation process at least one type has to be activated. This implies that in this setting there are at most $\text{nd}(G)$ rounds of the activation process. Note further that the two functions of our interest – majority and constant functions – both fulfil the assumption of Lemma 2. 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 1 ([11][7]).** 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 $O(p^{2.5p}\text{poly}(L))$ 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. Observe further that as the vertices in a type share all neighbors the only thing that matters is a 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 $C$ put 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 type by $T$ and we write $D \in N(C)$ if the two corresponding vertices in the type graph $T_G$ are joined by an edge.

**ILP formulation.**

\[
\begin{align*}
\text{minimize} \quad & \sum_{C \in T} x_C \\
\text{subject to} \quad & f'(C) \leq \sum_{D < C, D \in N(C)} n_D + \sum_{D > C, D \in N(C)} x_C \quad \forall C \in T \\
\text{where} \quad & 0 \leq x_C \leq n_C \quad \forall C \in T
\end{align*}
\]

As there are at most $t!$ orders of the set $[t]$, this implies that the Majority Target Set Selection problem can be solved in time $t!t^{O(t)}\text{poly}(n) = t^{O(t)}\text{poly}(n)$. Thus, we have proven Theorem 3.
4 Hardness Reductions

In this section we prove that Target Set Selection is \( W[1] \)-hard on graph of bounded neighborhood diversity and a general threshold function. We use an FPT-reduction from \( k \)-Multicolored Clique.

\( k \)-Multicolored Clique

\textit{Parameter:} \( k \)

\textit{Input:} \( k \)-partite graph \( G = (V_1 \cup \cdots \cup V_k, E) \), where \( V_a \) is an independent set for every \( a \in [k] \) and they are pairwise disjoint.

\textit{Task:} Find a clique of the size \( k \).

Let \( G \) be an input of \( k \)-Multicolored Clique. We refer to a set \( V_a \) as to a color class of \( G \) and to a set \( E_{ab} \) as to edges between color classes \( V_a \) and \( V_b \).

The problem is \( W[1] \)-hard \cite{6} even if every color class \( V_a \) has the same size and the number of edges between every \( V_a \) and \( V_b \) is the same. For easier notation during the reduction, we denote the size of arbitrary color class \( V_a \) by \( n+1 \) and the size of arbitrary set \( E_{ab} \) by \( m+1 \). We describe how to create from the graph \( G \) an instance \((G', f: V \rightarrow \mathbb{N}, k')\) of Target Set Selection such that:

1. The reduction runs in time \( \text{poly}(|G|) \).
2. The graph \( G \) has a clique of size \( k \) if and only if the graph \( G' \) has a target set of size \( k' \).
3. The neighborhood diversity of \( G \) is \( O(k^2) \). Moreover, all types of \( G' \) are independent set.

In the \( k \)-Multicolored Clique problem we need to select exactly one vertex from each color class \( V_a \) and exactly one edge from each set \( E_{ab} \). Moreover, we have to make certain that if \( \{u, v\} \in E_{ab} \) is a selected edge, then \( u \in V_a \) and \( v \in V_b \) are selected vertices. As the proof is quite long and technical we overview main ideas contained in the proof here.

**Overview of Proof of Theorem** \cite{2} We present a way of encoding a vertex \( v \) in a color class \( V_a \) of graph \( G \) by two numbers \( v \)-up and \( v \)-down with \( v \)-up + \( v \)-down = \( n \). We proceed with encoding of edges 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. Finally, we add an incidence check that has a vertex for each possible incidence between a vertex and an edge. Thus, we do this in both -up and -down parts. However, in this encoding all edges preceding the selected edge have their threshold also fulfilled – this happens in the -up part, while in the -down part all edges following the selected edge have their threshold fulfilled. The core of the proof relies on a fact that the threshold for the selected edge is fulfilled in both (-up and -down) parts if and only if the selected vertex is incident with it. It follows that there are only two possibilities – either \( m+1 \) or \( m+2 \) thresholds are fulfilled. Thus, we can test the incidence using threshold.

4.1 Proof of Theorem \cite{2}

Selection gadget. First, we describe gadgets of the graph \( G' \) for selecting vertices and edges of the graph \( G \). For an overall picture of the gadget please
refer to Figure 2. The gadget $L(s, t)$ is formed by two types $L$-down and $L$-up of equal size $s$ (the number $s$ will be determined later); we refer to these two types as selection part. For 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.

The last part of the gadget $L$ is formed by type $L$-guard of $t$ vertices connected to both types in the selection parts. The number $t$ has to be large enough (at least $s + 1$). For each vertex $v$ in $L$-guard type we set $f(v) = s$.

**Lemma 4.** Suppose there is a selection gadget $L(s, t)$ for $t > s$ in the input graph $G'$ of the problem of Target Set Selection. We claim that exactly $s$ vertices of the gadget $L$ are needed to be selected in the target set $S$ to activated the vertices in $L$-guard type. Moreover, these $s$ vertices have to be selected from the selection part of $L$.

**Proof.** Let $S' = V(L) \cap S$, i.e. vertices of the target set $S$ in the gadget $L$. First, suppose $|S'| < s$ or $|S'| = s$ and some vertex $u$ of $L$-guard is in $S'$. Since $t > s$, there is a vertex $v$ in $L$-guard type in the gadget $L$ such that $v \notin S'$. Let $V^p$ be vertices of the selection part of $L$. The vertex $v$ has neighbors only in $V^p$ and threshold of $v$ is $s$. Note that $|V^p \cap S'| < s$. Thus, at least one vertex $w \in V^p \setminus S'$ need to be activated during the activation process before the vertex $v$ is activated. However, $f(w) = \deg(w)$. Therefore, the vertex $w$ have to be activated after the vertex $v$ is activated. That is a contradiction and $|V^p \cap S'| \geq s$ must hold. When $S'$ contains $s$ vertices from the selection part of $L$, then it is easy to see that the all vertices in $L$-guard type are activated in the first round of the process. \qed
Numeration of Vertices and Edges. Now, we informally describe how we use selection gadget. We numerate the vertices in each color class arbitrarily. Let $V_a = \{v_0, \ldots, v_n\}$. By Lemma 4, we can encode selecting vertices and edges of the graph $G$ to the multicolor clique. For every color class $V_a$ we create a selection gadget $L_a(n + 1, n)$. We select a vertex $v_i \in V_a$ to the multicolor clique if $i$ vertices in the $L_a$-up type of the gadget $L_a$ are selected into the target set (and $n - i$ vertices in the $L_a$-down type are selected into the target set).

The selection of edges is similar, however complicated. Let $q \in \mathbb{N}$ and $E_{ab} = \{e_0, \ldots, e_m\}$. For every set $E_{ab}$ we create a selection gadget $L_{ab}(qm + 1, qm)$. We select an edge $e_j \in E_{ab}$ to the multicolor clique if $q_j$ vertices in the $L_{ab}$-up type of the gadget $L_{ab}$ are selected into the target set (and $q(m - j)$ vertices in the $L_{ab}$-down are selected into the target set). Suppose $s$ vertices in the $L_{ab}$-up type are selected into the target set. If $s$ is not divisible by $q$, then it is invalid selection. We introduce new gadget such that $s$ has to be divisible by $q$.

Multiple Gadget. A multiple gadget $M(q, t', s, t)$ contains a selection gadget $L(qs, t)$ and 3 other types $M$-up, $M$-down of $s$ vertices and $M$-guard of $t'$ vertices. The type $M$-up is connected to the type $L$-up and the type $M$-down is connected to the type $L$-down. The type $M$-guard is connected to the types $M$-up and $M$-down. Still, the rest of graph $G'$ is connected only to types $L$-up and $L$-down. Let $\{u_1, \ldots, u_s\}$, $\{w_1, \ldots, w_s\}$ be vertices in $M$-up type, $M$-down type respectively. We set thresholds $f(u_i) = f(w_i) = qi$. For each vertex $v$ in $M$-guard we set $f(v) = s$. For an example of multiple gadget see Figure 3.

Lemma 5. Suppose there is a multiple gadget $M(q, t', s, t)$ for $t > s, t' \geq qs$ in the input graph $G'$ of TARGET SET SELECTION. Let $L$ be a selection gadget in $M$. We claim that exactly $qs$ vertices of the gadget $L$ are needed to be selected in the target set $S$ to activate the types $L$-guard, $M$-up, $M$-down and $M$-guard.
Moreover, these \(qs\) vertices have to be selected from the selection part of \(L\) and the numbers of vertices selected in \(L\)-up and \(L\)-down types are divisible by \(q\).

**Proof.** By Lemma [4] we know that \(qs\) selected vertices in the types \(L\)-up and \(L\)-down are needed to activate \(L\)-guard type. Suppose there is \(z\) vertices in the \(L^{up}\) type selected into a target set and \(k\) is not divisible by \(q\). It follows that there is \(qs - z\) selected vertices in \(L^{down}\). Thus, \(z = qa + b, b \neq 0\) and \(qs - z = q(s - a) - b\). Let \(\{u_1, \ldots, u_s\}, \{w_1, \ldots, w_s\}\) be vertices in the \(M\)-up type, in the \(M\)-down type respectively. Recall that \(f(u_i) = f(w_i) = qi\). Thus, vertices \(u_1, \ldots, u_s\) and \(w_1, \ldots, w_{s-a-1}\) are activated in the first round of the activation process.

We claim that no other vertices in gadget \(M\) would be activated during the process. Vertices in \(M\)-guard type have only \(s - 1\) activated vertices among their neighbors and have thresholds \(s\). Vertices in \(L\)-up and \(L\)-down have thresholds their degrees. Thus, they have be activated after all vertices in \(M\)-up and \(M\)-down are activated. Vertices \(u_{a+1}, \ldots, u_s\) in \(M\)-up type and \(w_{s-a}, \ldots, w_s\) cannot be activated unless some of their neighbors are activated.

Now suppose that \(b = 0\), i.e. \(z = qa\) and \(qs - z = q(s - a)\). Vertices \(u_1, \ldots, u_a\) and \(w_1, \ldots, w_{s-a}\) are activated in the first round. All vertices in the \(M\)-guard type are activated in the second round because they have \(s\) activated vertices among their neighbours. Recall that the maximum threshold in the \(M\)-up and the \(M\)-down type is \(qs\). Since \(t' \geq qs\), every vertex in the types \(M\)-up and \(M\)-down has at least \(qs\) activated vertices among its neighbours. Therefore, all vertices in the types \(M\)-up and \(M\)-down are activated in the third round. \(\square\)

**Incident 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 \(v \in V_a\) and \(e \in E_{ab}\) are selected vertex and edge to multicolor clique, then \(v \in e\). We create \(L_a(n, n + 1)\) selection gadget for a color class \(V_a\). We set the number \(q\) to \(n^2\) and create a multiple gadget \(M_{ab}(n^2, n^2m, m, m + 1)\) (with selection gadget \(L_{ab}\)) for a set \(E_{ab}\). We join gadgets \(L_a\) and \(M_{ab}\) through an incident gadget \(I_{a,ab}\). See Figure 4 for better understanding how the incident gadget is connected to the selection and multiple gadgets. The incident gadget \(I_{a,ab}\) has three types \(I_{a,ab}\)-up and \(I_{a,ab}\)-down of \(m + 1\) vertices and \(I_{a,ab}\)-guard of \(n + n^2m\) vertices. We connect the \(I_{a,ab}\)-guard type to the types \(I_{a,ab}\)-up and \(I_{a,ab}\)-down. Furthermore, we connect the type \(I_{a,ab}\)-up to the types \(L_a\)-up and \(L_{ab}\)-up. Similarly, we connect the type \(I_{a,ab}\)-down to the types \(L_a\)-down and \(L_{ab}\)-down.

We set thresholds of all vertices in the \(I_{a,ab}\)-guard type to \(m + 2\). Recall there is \(m + 1\) edges in the set \(E_{ab}\). Thus, we can associate edges in \(E_{ab}\) with vertices in \(I_{a,ab}\)-up (\(I_{a,ab}\)-down respectively) one-to-one. I.e., \(V(I_{a,ab}\text{-up}) = \{u_e \mid e \in E_{ab}\}\) and \(V(I_{a,ab}\text{-down}) = \{w_e \mid e \in E_{ab}\}\). Let \(v_i \in V_a, e_j \in E_{ab}\) 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 \(L_a\)-up type and \(n^2j\) vertices in \(L_{ab}\)-up type into a target set. We set threshold of \(w_j\) to \(i + n^2j\) and threshold of \(w_j\) to the ”opposite” value \(n - i + n^2(m - j)\).
Fig. 4. An overview of the reduction. Number inside a type is the number of vertices of the type. Beneath each type the threshold is show in case it is the same for each vertex in the type.

Since we set the coefficient $q$ to $n^2$, for each edge $e_j \in E_{ab}$ and each vertex $v_i \in V_a$ the sum $i + n^2 j$ is unique. Thus, every vertex in $I_{a:ab}$-up ($I_{a:ab}$-down) 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):

1. For every color class $V_a$ we create a selection gadget $L_a$.
2. For every edge set $E_{ab}$ we create a multiple gadget $M_{ab}$.
3. We join the gadgets $L_a$ and $M_{ab}$ by an incident gadget $I_{a:ab}$ (gadgets $L_b$ and $M_{ab}$ are joint by a gadget $I_{b:ab}$).

It is easy to see the following observations by constructions of $G'$.

**Observation 7** The graph $G'$ has polynomial size in the size of the graph $G$.

**Observation 8** Neighborhood diversity of the graph $G'$ is $O(k^2)$.

To finish the instance of TARGET SET SELECTION we set budget for target set

$$k' = kn + \binom{k}{2} n^2 m.$$

**Theorem 9.** If the graph $G$ contains a clique of size $k$, then $G'$ with the threshold function $f$ contains a target set of size $k'$.
Theorem 10. If the graph $G'$ with the threshold function $f$ contains a target set of size $k'$, then $G$ contains a clique of size $k$.

Proof. Let $S$ be a target set of the graph $G$ of size $k'$. There are $k$ selection gadgets $L(n, n+1)$ in $G'$. By Lemma 4 the set $S$ has to contain at least $n$ vertices in selection part of every gadget $L(n, n+1)$. There are also $\binom{k}{2}$ selection gadgets $L(n^2 m, n^2 m + 1)$ in multiple gadgets in $G'$. By Lemma 5 the set $S$ has to contain at least $n^2 m$ vertices in selection part of every gadget $L(n^2 m, n^2 m + 1)$. Since $|S| = k' = kn + \binom{k}{2}n^2 m$, there is not any other vertex in $S$. 

Proof. Let $K$ be a $k$-clique in the graph $G$. We construct a set $S \subseteq V(G')$. Let $v_i \in V(K) \cap V_a$. We add $i$ vertices in $L_a$-up type and $n - i$ in $L_a$-down type into the set $S$. Let $e_{ij} \in E(K) \cap E_{ab}$. For the set $E_{ab}$ we have a multiple gadget $M_{ab}$ and there is a selection gadget $L_{ab}$ inside $M_{ab}$. We add $n^2 j$ vertices in the $L_{ab}$-up type and $n^2 (m - j)$ vertices in the $L_{ab}$-down into the set $S$. We have $n$ vertices in $S$ for every color class $V_a$ and $n^2 m$ vertices in $S$ for every edge set $E_{ab}$. Thus,

$$|S| = kn + \frac{k}{2}n^2 m = k'.$$

We claim that the set $S$ is a target set. We analyze the selection gadget $L_a$, the multiple gadget $M_{ab}$ (with the $L_ab$ selection gadget) and the incident gadget $I_{ab}$. All vertices in the types $L_a$-guard and $L_{ab}$-guard are activated in the first round (see proof of Lemma 4). All vertices in the types $M_{ab}$-down, $M_{ab}$-up and $M_{ab}$-guard are activated during first three rounds – for details see proof of Lemma 5.

Recall $V(I_{a:ab}$-up) = $\{ u_{\ell} \mid e_{\ell} \in E_{ab} \}$ and $V(I_{a:ab}$-down) = $\{ w_{\ell} \mid e_{\ell} \in E_{ab} \}$. Threshold of $u_{\ell} \in V(I_{a:ab}$-up) is $n^2 \ell + \ell'$ for some $\ell' \in \{0, \ldots, n \}$. There are $n^2 j + i$ vertices activated in the types $L_{ab}$-up and $L_a$-up. Vertices $u_0, \ldots, u_{j-1}$ are activated in the first round because their thresholds are strictly smaller than $n^2 j$. The threshold of $u_j$ is $n^2 j + i$ because this vertex corresponds to the incidence $v_i \in e_j$. Thus, the vertex $u_j$ is activated in the first round as well. Vertices $u_{j+1}, \ldots, u_m$ have thresholds bigger than $n^2 (j + 1)$ and cannot be activated in the first round. By the same analysis we get that vertices $w_j, \ldots, w_m$ in the $I_{a:ab}$-down type are activated in the first round.

In the first round there are $m + 2$ activated vertices in the types $I_{a:ab}$-up and $I_{a:ab}$-down. All vertices in the $I_{a:ab}$-guard type are activated in the second round because they have threshold $m + 2$. Maximum threshold in the $I_{a:ab}$-up ($I_{a:ab}$-down) type is $n + n^2 m$. All vertices in the types $I_{a:ab}$-up and $I_{a:ab}$-down are activated in the third round because there are $n + n^2 m$ activated neighbors in $I_{a:ab}$-guard type.

All vertices outside the types $L_a$-up, $L_a$-down, $L_{ab}$-up and $L_{ab}$-down are activated during three rounds. Let $U$ be a set of vertices which are not activated during three rounds. Note that $U$ is an independent set and for every $u \in U$ holds that $f(u) = \deg(u)$. Therefore, vertices in $U$ are activated in the fourth round. 

□
Now, for every \( V_a \) and \( E_{ab} \) we select a vertex (or an edge, respectively). We select a vertex \( v_i \in V_a \) if \( |V(L_a-) \cap S| = i \). We select an edge \( e_j \in E_{ab} \) if \( |V(L_{ab}-) \cap S| = q_j \). By Lemma 4 and 5 we know the selection is correct. We claim that if \( v_i \in V_a \) is the selected vertex and \( e_j \in E_{ab} \) is the selected edge, then \( v_i \in e_j \).

For a contradiction suppose \( v_i \not\in e_j \). We analyze the incident gadget \( I_{a:ab} \). Let \( V(I_{a:ab}:up) = \{u_0, \ldots, u_m\} \) and \( V(I_{a:ab}:down) = \{w_0, \ldots, w_m\} \). Vertices in the type \( I_{a:ab}:up \) have \( i + n^2j \) active neighbors. Vertices in the type \( I_{a:ab}:down \) have \( n - i + n^2(m - j) \) active neighbors. As we say in the proof of Theorem 9 vertices \( u_0, \ldots, u_{j-1} \) and \( w_{j+1}, \ldots, w_m \) are activated in the first round and vertices \( u_{j+1}, \ldots, u_m \) and \( w_0, \ldots, w_{j-1} \) are not activated.

It remains to analyze the vertices \( u_j \) and \( w_j \). Suppose \( u_j \) is activated in the first round. Thus, \( f(u_j) = i' + n^2j < i + n^2j \). Note that \( i' < i \) because we suppose \( v_i \not\in e_j \). For threshold of \( w_j \) holds

\[
f(w_j) = n - i' + n^2(m - j) > n - i + n^2(m - j).
\]

Since the vertex \( w_j \) has \( n - i + n^2(m - j) \) activated neighbors, the vertex \( w_j \) cannot be activated in the first round. Thus, at least one of the vertices \( u_j, w_j \) is not activated in the first round.

Any vertex of the type \( I_{a:ab}:guard \) cannot be activated in the first round because they have threshold \( m + 2 \) and they have at most \( m + 1 \) activated neighbors. Vertices of the type \( I_{a:ab}:guard \) have to be activated after some other vertices in the types \( I_{a:ab}:up \) or \( I_{a:ab}:down \) are activated. However, there is not any new activated vertex in the neighborhood of the types \( I_{a:ab}:up \) and \( I_{a:ab}:down \). The activation process does not activate all vertices in \( V(G') \). Therefore, \( S \) is not a target set, which is a contradiction.

\[\square\]

Theorem 2 is a corollary of Theorem 9, 10 and Observation 7, 8.

**4.2 Proof of Theorem 4**

**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. However, for vertices with threshold deg one has to “double” the number of vertices in the neighborhood and make sure that no such vertex is activated before all with threshold deg. Finally, one has to deal with types having different thresholds for each of its vertices. Here we exploit the property of the previous proof – that these vertices naturally come in pairs and that it is possible to replace each of these vertices by a collection of cliques. This ensures that even if the neighborhood is the same some vertices get activated and some not.
5 Conclusions

We have generalized ideas of previous works \cite{2,15} for the Target Set Selection problem. The presented results give new methods for solving and showing \text{W}[1]-hardness result. In particular, only few problems are known to be \text{W}[1]-hard when parameterized by neighborhood diversity – which is the case for the Target Set Selection problem.

Thus, we would like to address several open problems regarding structural parameterizations of the Target Set Selection problem. Determine parameterized complexity of

- the Majority Target Set Selection problem parameterized by cluster vertex deletion number number \cite{5};
- the Target Set Selection problem parameterized by the modular-width and the threshold upper-bound \( t \) (that is, \( f(v) \leq t \) for each vertex \( v \));
- the Target Set Selection problem parameterized by twin cover number;
- the Target Set Selection problem parameterized by the distance to clique \cite{5}.

References

1. Eyal Ackerman, Oren Ben-Zwi, and Guy Wolfovitz. Combinatorial model and bounds for target set selection. Theoretical Computer Science, 411(44):4017 – 4022, 2010.
2. Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. Discrete Optimization, 8(1):87 – 96, 2011. Parameterized Complexity of Discrete Optimization.
3. Ning Chen. On the approximability of influence in social networks. SIAM Journal on Discrete Mathematics, 23(3):1400–1415, 2009.
4. 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.
5. 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.
6. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Parameterized Algorithms. Springer, 2015.
7. A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica, 7(1):49–65, 1987.
8. J. Gajarský, M. Lampis, and S. Ordyniak. Parameterized algorithms for modular-width. In IPEC 2013, Revised Selected Papers, pages 163–176, 2013.
9. R. Ganian. Twin-cover: Beyond vertex cover in parameterized algorithmics. In IPEC 2011, Revised Selected Papers, pages 259–271, 2011.
10. Falk Hüffner, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. Theory Comput. Syst., 47(1):196–217, 2010.
11. H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

12. David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

13. M. Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

14. Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014.

15. 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.

16. David Peleg. Local majorities, coalitions and monopolies in graphs: A review. *Theor. Comput. Sci.*, 282(2):231–257, 2002.

17. M. Tedder, D. G. Corneil, M. Habib, and C. Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *ICALP 2008*, pages 634–645, 2008.