Target Set Selection Parameterized by Clique-Width and Maximum Threshold

Tim A. Hartmann(✉)

Lehrstuhl für Informatik 1, RWTH Aachen University, Aachen, Germany
hartmann@algo.rwth-aachen.de

Abstract. The Target Set Selection problem takes as an input a graph G and a non-negative integer threshold $\mathsf{thr}(v)$ for every vertex v. A vertex v can get active as soon as at least $\mathsf{thr}(v)$ of its neighbors have been activated. The objective is to select a smallest possible initial set of vertices, the target set, whose activation eventually leads to the activation of all vertices in the graph.

We show that Target Set Selection is in FPT when parameterized with the combined parameters clique-width of the graph and the maximum threshold value. This generalizes all previous FPT-membership results for the parameterization by maximum threshold, and thereby solves an open question from the literature. We stress that the time complexity of our algorithm is surprisingly well-behaved and grows only single-exponentially in the parameters.

1 Introduction

The Target Set Selection problem (TSS) suits to model irreversible propagation of all sorts of conditions or information in a network. This may be for example a word-of-mouth-effect, disease spreading or fault influence in distributed systems [15]. The input is an undirected graph G and a non-negative integer threshold $\mathsf{thr}(v)$ for every vertex v. The task is to select a smallest possible set S of initially active vertices, the target set, whose activation eventually leads to the activation of all vertices in the graph. A vertex v can become active as soon as at least $\mathsf{thr}(v)$ of its neighbors have been activated.

Our view on the activation of a vertex is that it is *allowed* to become active if enough neighbors are active before, in contrast to that it is *obligated* to get active as soon as possible. We ask for a smallest possible set S, the target set, and a permutation of the vertices π, which is the ordering in which the vertices get active. Then, for every non-target set vertex v, to assure its activation we require that at least threshold $\mathsf{thr}(v)$ many neighbors of v are ordered before v. In particular, our permutation may order the target set vertices S not at the beginning. This definition is more robust towards re-orderings of the permutation of vertices. We can re-order the permutation and not have to bother that for example the target set no longer consists of the very first vertices of the ordering. In the literature the problem is commonly defined via rounds of activations that

A M. Tjoa et al. (Eds.): SOFSEM 2018, LNCS 10706, pp. 137–149, 2018.
https://doi.org/10.1007/978-3-319-73117-9_10

define sets of active vertices for each round. Our definition is equivalent while being much more convenient for our techniques.

Target Set Selection
Input: An undirected graph G, a non-negative threshold for every vertex $\mathsf{thr} : V(G) \to \mathbb{N}$, and $k \in \mathbb{N}$.
Question: Is there a set of vertics $S \subseteq V(G)$ of size at most k and a permutation of the vertices $\pi : V(G) \to [|V(G)|]$ such that for every vertex $v \in V(G) \setminus S$ we have $|\{u \in \mathsf{N}_G(v) \mid \pi(u) < \pi(v)\}| \geq \mathsf{thr}(v)$?

The problem was first introduced by Kempe et al. [14]. It proves to be computationally extremely difficult. It is NP-hard even for the restriction to split-graphs of diameter two [15]. Chen showed that minimizing the size of the target set is APX-hard [4]. More recently, Bazgan et al. showed that for every functions f and ρ this problem cannot be approximated within a factor of $\rho(k)$ in $f(k) \cdot n^{\mathcal{O}(1)}$ time [1]. The parameterized complexity studies focus on the original problem and two variants that limit the allowed thresholds. These are *constant thresholds*, where all thresholds are at most a constant $t_{\max}$, and *majority thresholds*, where a vertex can get active as soon as at least the majority of its neighborhood is active before. The general TSS is W[1]-hard for each of the parameterization, "distance to cluster," [5] "distance to forest" and pathwidth [15]. The strongest positive FPT-membership results for constant thresholds are the parameterization by treewidth [2], the parameterization by "distance to cluster" [5], and the parameterization by neighborhood diversity [11]. There are a lot more parameterized complexity results for these three variants of TSS [5,15]. Further, Cicalese et al. study a variant of TSS which asks if a set of vertices A can be activated in a given number of activation rounds [6]. They give a polynomial time algorithm when the number of activation rounds and the clique-width of the input graph are constant. Their exponential dependency on the clique-width is unlikely to be improved, as even TSS for one activation round is W[1]-hard with respect to the treewidth [3]. For a more extend introduction to the history of the problem as well as other algorithmic aspects and similar models see for example [5,15].

Dvořák et al. raised the question of the complexity of the parameterization by the modular-width [11]. The structural graph parameter modular-width was introduced by Gajarský et al. [13]. We give a positive answer by showing FPT-membership for a more general question. We consider the clique-width which is upper bounded by the parameters modular-width and treewidth [7], and by further common structural parameters for which the parametrized complexity of TSS was open. Thereby, we generalize all positive FPT-memberships results for TSS with constant thresholds. Further, our result does not rely on the maximum threshold $t_{\max}$ being a constant, but allows that $t_{\max}$ is a parameter. Moreover, the time complexity of our algorithm behaves surprisingly well and grows only single-exponentially in the parameters clique-width and maximum threshold.

A related result is that TSS is in FPTwhen parameterized by treewidth and maximum threshold, by Ben-Zwi et al. [2]. They use a dynamic program that works along the bags of a computed tree decomposition. They fix the local ordering in which the vertices of the currently observed bag get active. Our approach also uses such an recursive approach, while working on a computed ℓ-expression. Informally, an ℓ-expression is a tree-decomposition in the context of clique-width. Such an ℓ-expression f uses three types of recursive operations that work on labeled vertices using at most ℓ different labels. Analogously to the approach for a tree decomposition, for every subexpression a current state fixes a part of the global ordering of the vertices.

However, the described vertices of a current subexpression is not bounded by our parameters. Our algorithm has to remember an ordering of a limited number of vertices and further has to address these vertices indirectly. Crucial for the activation of a vertex is its threshold and neighborhood. However, we cannot address the neighborhood even for vertices of currently equal label and threshold since they can have very different neighborhoods as subexpression may reveal. Consequently, our approach explores the ℓ-expression top down, and fixes an ordering of the important vertices of the up to now described graph. The up to now encountered operations define a common neighborhood for all vertices of a fixed label. This is because for every outer operations, vertices of the same label behave equally. Thus, our local ordering indirectly references the vertices solely by their label and threshold.

Further, vertices of the same label that occur late enough in a global ordering behave equally. There is only one type of edge operation of ℓ-expression, namely $\eta_{\alpha,\beta}$ adding all edges between vertices of some labels α and β. There, for a vertex v of label α we have to account the contribution to the activation of v due to vertices of label β. Only the first $\mathsf{thr}(v) \leq t_{\max}$ active vertices of label β are important. If the activation of v is between the activation of the first $t_{\max}$ of label β, we fix their relative positioning in our local ordering. Otherwise, the activation of v does not differ from other late vertices of label α.

However, we need to guarantee that a vertex v of label α that is not referenced by our local ordering is indeed ordered late enough. That is, the first $t_{\max}$ vertices of label β occur before vertex v. We denote such a global ordering as nice to the current subexpression $\eta_{\alpha,\beta}f'$. It is possible to modify any valid global ordering to be nice to all subexpressions. We extend our local ordering to also include the $(t_{\max}+1)$-st vertex of every label. Then, whether the underlying global ordering π is nice, is reflected in our local ordering. Therefore, we can restrict our algorithm to consider nice global orderings only.

The resulting procedure for our algorithm at each operation of the given ℓ-expression then is as follows. For a current edge operation $\eta_{\alpha,\beta}$, for each vertex v we simply have to adjust the number of neighbors contributing to the activation of v according to our fixed local ordering. We remember this contribution as the *activation from outside.* For a current operation that combines two subgraphs, consider the unknown partition of the vertices fixed by the local ordering in either subgraph. In that case, the algorithm tries all possibilities. The approach for the operation that re-labels a label is very similar. For every subexpression, the number of possible states is single-exponentially bounded by our parameters, which yields to an overall FPT-runtime.

Theorem 1. *Let $t_{\max}, \ell \in \mathbb{N}$. There is an algorithm that, given a graph G, a threshold for each vertex $\mathsf{thr} : V(G) \to [0, t_{\max}]$ and an ℓ-expression f of G, computes the minimal size of a target set in time $\mathcal{O}(\ell^{3\ell t} \cdot t^{\ell(4t+1)} \cdot |f|)$, where $t := t_{\max} + 1$ and $|f|$ is the length of f.*

An easy upper bound for the length of the ℓ-expression f is $|V(G)|^2$. Further, one can obtain a minimum target set, and not only its size, by tracking such sets throughout our dynamic program.

Oum gave an algorithm that either outputs an $(8^\ell - 1)$-expression of graph G or confirms that the clique-width of G is larger than ℓ, and that runs in time $\mathcal{O}(g(\ell) \cdot |V(G)|^3)$, where $g(\ell)$ only depends on the clique-width ℓ [17]. Combined with the algorithm of Theorem 1 it follows that TSS parameterized by the clique-width and the maximum threshold is in FPT.

Corollary 1. Target Set Selection *is in FPT with respect to the combined parameters clique-width of the given graph and the maximum threshold.*

Following the preliminaries in Sect. 2, we prove Theorem 1 in Sect. 3. We conclude in Sect. 4. Due to space constraints, we omit some proofs or only give a proof sketch. For the full proof, we refer to an online version at https://arxiv.org/abs/1710.00635.

2 Preliminaries

For integers $i < j$, let $[i] := \{1, 2, \ldots, i\}$ and $[i, j] := \{i, (i+1), \ldots, j\}$. For a list (or vector) A, we describe the i-th element as $A[i]$.

All our graphs are simple, finite and undirected. For a graph G, we denote by $V(G)$ its set of vertices. We use $\mathsf{N}_G(v)$ as the neighborhood of vertex $v \in V(G)$. Usually we consider graphs with thresholds for each vertex $\mathsf{thr} : V(G) \to [0, t_{\max}]$ which are at most a constant $t_{\max}$, and assume that its thresholds thr and $t_{\max}$ are given, if needed.

In this work, we consider parameterized complexity. For an introduction see for example [9,10,12,16]. For a graph class, for example clusters (the disjoint union of cliques), the parameter "distance to cluster" is the minimal number of vertices one needs to delete from the input graph in order to obtain a cluster.

The clique-width $\mathsf{cw}(G)$ of a graph G was introduced in [8]. A graph has clique-width at most $\ell \in \mathbb{N}$, if it can be constructed by an ℓ-expression that uses four types of operations and a labeling of the vertices of at most ℓ labels, as we describe in the following. Let $\mathsf{labels}(f)$ be the set of labels used by f. To avoid confusion with thresholds, we use small Greek letters α, β, γ for the labels. An ℓ-expression defines a graph $G(f)$ with labels per vertex $\mathsf{lab}_G : V(G) \to \mathsf{labels}(f)$. The graph $G(f)$ is recursively defined as

- $G(v(\alpha))$, a single vertex v of label $\alpha \in \mathsf{labels}(f)$,
- $G(f_1 \oplus f_2)$, the disjoint union of $G(f_1)$ and $G(f_2)$ for ℓ-expressions f_1, f_2,
- $G(\eta_{\alpha,\beta} f')$, the graph $G(f')$ where there is an **e**dge between every vertex of label α and every vertex of label β, for ℓ-expression f', and
- $G(\rho_{\alpha\to\beta} f')$, the graph $G(f')$ where all vertices of label α are **r**e-labeled to label β, for ℓ-expression f'.

The subexpressions of f are all expressions f_1, f_2, f' used in the recursive definition of f. Especially f is a subexpression of f. We drop the $G(\cdot)$ when using $G(f)$ as a nested term. For example, instead of $V(G(f))$, we simply write $V(f)$. Further, we also refrain from specifying the set of labels $\mathsf{labels}(f)$ if it is clear from the context.

An ℓ-expression is irredundant if for every subexpression $\eta_{\alpha,\beta} f'$ the graph $G(\eta_{\alpha,\beta} f')$ has no edge between vertices of label α and β. We assume that the given ℓ-expression is irredundant, which we can assure by a simple preprocessing step [8].

3 Dynamic Program

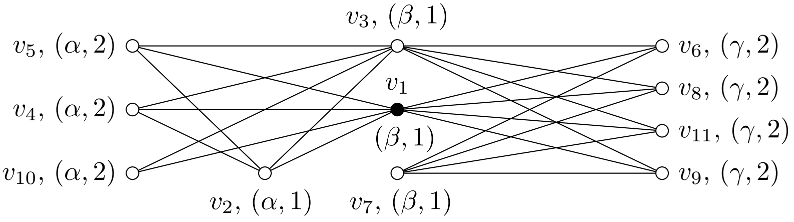
A good way to convince someone that a graph G with thresholds has a target set of size at most k is to state a complete ordering in which the vertices get active. We denote this permutation of the vertices as a *global ordering* $\pi : V(G) \to [|V(G)|]$. We say that π is k-activating for graph G if there is a k-vertex set $S \subseteq V(G)$, the target set, such that for every other vertex v the neighbors of v that are ordered before v outnumber the threshold $\mathsf{thr}(v)$.

Definition 1. *A* global ordering *of a graph G is a permutation of the vertices $\pi : V(G) \to [|V(G)|]$. Further, π is k-activating (for G) if there is a k-vertex set $S \subseteq V(G)$ such that for every vertex $v \in V(G) \setminus S$ we have*

$$\pi_G^{<}(v) \;:=\; \big|\{u \in \mathsf{N}_G(v) \;\big|\; \pi(u) < \pi(v)\}\big| \;\geq\; \mathsf{thr}(v).$$

Graph G has a target set of size k if there is a global ordering π such that π is k-activating for G.

Example 1. The following graph G has global ordering $\pi : v_i \mapsto i$, which is 1-activating (for $S = \{v_1\}$). Further, $f = \eta_{\beta,\gamma} f' = \eta_{\beta,\gamma}(v_6(\gamma) \oplus v_8(\gamma) \oplus v_{11}(\gamma) \oplus v_9(\gamma) \oplus v_7(\beta) \oplus \rho_{\gamma\to\alpha}\eta_{\beta,\gamma}(v_{10}(\gamma) \oplus \rho_{\gamma\to\alpha}\eta_{\alpha,\beta}\eta_{\alpha,\gamma}\eta_{\beta,\gamma}(v_2(\gamma) \oplus v_1(\beta) \oplus v_3(\beta) \oplus v_4(\alpha) \oplus v_5(\alpha))))$ is a 3-expression of G. For each vertex, the label among $\{\alpha, \beta, \gamma\}$ and threshold at most $t_{\max} = 2$ is given as a tuple.



For later examples, let $A := \big((\beta,1),(\alpha,1),(\beta,1),(\alpha,2),(\alpha,2),(\gamma,2),(\beta,1),(\gamma,2),(\gamma,2)\big)$, and further $\eta_{\alpha,\beta}f'$, G and π be as defined here.

An ℓ-expression f describes a graph $G(f)$ with three types of recursive operations that rely on ℓ different labels assigned to the vertices. We formulate a dynamic program over the subexpressions of f. At a current subexpression f, a *state* fixes a part of a global ordering π. Whether such a state is a part of a k-activating global ordering, is verified by considering the subexpressions with suitable states.

In order to obtain the desired FPT-runtime, we may only work with states that fix an ordering of a number of vertices bounded by our parameters, which are maximum threshold $t_{\max}$ and clique-width ℓ. However, the number of all vertices described by a current subexpression is not bounded by our parameters. Our algorithm thus can only remember an ordering of a limited number of vertices and further cannot address these vertices directly. We identify the important verices and a suitable way to remember them. Crucial for the activation of a vertex is its threshold and neighborhood. Our local ordering can very well remember the threshold of vertices. However, it cannot address the neighborhood even for vertices of currently equal label and threshold since they can have very different neighborhoods as subexpression may reveal.

Consequently, our approach explores the given ℓ-expression top down, and fixes an ordering of the important vertices of the graph described by the up to now seen part of the ℓ-expression. The up to now seen operations define a common neighborhood for all vertices of a fixed label. This is because for every outer operation, two vertices of equal label behave equally. Thus, our local ordering can indirectly reference the vertices solely by their label and threshold.

Now, let us identify the vertices whose relative ordering is crucial. We can observe that vertices of the same label that occur late enough in a global ordering behave equally. An ℓ-expression has only one type of operation that adds edges, namely $\eta_{\alpha,\beta}$ for some labels α and β, which adds all edges between vertices of labels α and β. There, for a vertex v of label α we have to account for the contribution to the activation of v by the vertices of label β. Only the first $\mathsf{thr}(v) \leq t_{\max}$ vertices of label β of the global ordering π are important. Consequently, if π orders v somewhere between the first $t_{\max}$ vertices of label β, the local ordering fixes the ordering of v relatively to those first vertices of label β as well. If π orders v after the first $t_{\max}$ of label β, we can neglect its exact ordering. This is because the number of neighbors of label β that contribute to its activation do not differ from other such late vertices of label α. Our plan therefore is that the local ordering fixes the relative positioning of these crucial first $t_{\max}$ vertices of every label.

Doing so, we need to guarantee that a vertex v of label α that is not referenced by our local ordering is indeed ordered late enough. That is, the first $t_{\max}$ vertices of label β occur before vertex v. In particular, the first $t_{\max}$ vertices of label β are ordered before the $(t_{\max}+1)$-st of label α. Then, given that v is not referenced by our local ordering, there are at least $t_{\max}$ of label β ordered before, or if there are not even as many of label β, accordingly less. We denote such an ordering as nice to the current subexpression $\eta_{\alpha,\beta} f'$. It is possible to modify any valid global ordering such that it is nice to every subexpression. Therefore, our algorithm may only consider nice global orderings. We extend our local ordering to also include the $(t_{\max} + 1)$-st vertex of every label. Then, whether the underlying global ordering π is nice to a current expression $\eta_{\alpha,\beta} f'$, is reflected in our local ordering. Our algorithm may then ignore states with such not nice local orderings.

We define the local ordering A for a current ℓ-expression f that fixes the relative ordering of the first $(t_{\max} + 1)$ activate vertices for each label α (or if there are not even as many vertices of label α, accordingly less), which we denote by t^α. We indirectly remember a vertex v by fixing the label and threshold of v. For technical reasons, we define a local ordering as possibly incomplete. Our algorithm only considers complete local orderings.

Definition 2. *Let G be a graph with labels $\mathsf{lab} : V(G) \to \mathsf{labels}(G)$. For label α, let $t^\alpha(G) := \min\{t_{\max}(G) + 1,\ |\{v \in V(G) \mid \mathsf{lab}(v) = \alpha\}|\}$. A* local ordering *A of G is a list of tuples of label and threshold $(\alpha, a) \in \mathsf{labels}(G) \times [0, t_{\max}(G)]$ such that for every label α there are at most t^α tuples of label α; and A is* complete *if, for every label α, there are exactly t^α tuples of label α.*

The local ordering A is our limited view on a global ordering π. Let $\mathsf{condense}(\pi)$ be the ordered list of vertices consisting of the first t^α vertices of each label α. A global ordering π *extends* A if the tuples of label and threshold of $\mathsf{condense}(\pi)$ are equal to A. As a technical tool, we also define $\mathsf{condense}(\pi, A)$ as the first ordered vertices consisting of each label α, such that the number of vertices labeled α is equal to as there are in A.

Definition 3. *Let graph G have global ordering π. Consider the list of vertices according to the global ordering $\pi^{-1}(1), \ldots, \pi^{-1}(|V(G)|)$. For every label α, remove all vertices of label α but the first t^α vertices of label α. Then, the resulting list is $\mathsf{condense}(\pi)$. Global ordering π* extends *a local ordering A (for G) if the list tuples of label and threshold of $\mathsf{condense}(\pi)$ is equal to A.*

Let $\mathsf{condense}(\pi, A)$ be the remaining list, after, for every label α, removing all vertices of label α but the first $|\{i \mid \mathsf{lab}(A[i]) = \alpha\}|$ of label α.

Example 2. We have $t^\alpha, t^\beta, t^\gamma = 3$ and A is a complete local ordering of G. Further, $\mathsf{condense}(\pi) = \mathsf{condense}(\pi, A) = (v_1, \ldots, v_9)$, whose list of tuples of label and threshold is equal to A. Thus, A extends π. Let incomplete local ordering A^* contain only one tuple per label. Then, $\mathsf{condense}(\pi, A^*)$ is the list of vertices (v_1, v_2, v_6). The list of tuples of label and threshold is equal to A^*.

For an edge operation $\eta_{\alpha,\beta}$, which adds all edges between vertices of two distinct labels, we simply have to adjust the number of neighbors contributing to an activation of a vertex according to our fixed local ordering. We remember this contribution as the *activation from outside*. The mapping afo maps to a value $[0, t_{\max}]$ for each position of the local ordering A, as well as maps to a value for each label. That way we have a value for every vertex indirectly referenced by

A. Further, there is a value for every vertex v not referenced by A, which we identify via the label of v.

A *state* of a current subgraph $G(f)$ is a tuple consisting of a local ordering A and an activation from outside afo. To reference the activation from outside for a concrete vertex v we define $A^\pi(v)$ such that $\mathsf{afo}(A^\pi(v))$ is the activation from outside for v. Thus, $A^\pi(v)$ maps v to its according position in A if it exists and otherwise to the label of v. A global ordering π is k-activating for a state (A, afo) of G if it is k-activating for G while supported by the activation from outside afo.

Definition 4. *Let f be an ℓ-expression, and graph $G(f)$ have local ordering A. An* **a**ctivation **f**rom **o**utside *for A is a mapping* $\mathsf{afo} : [|A|] \cup \mathsf{labels}(f) \to [0, t_{\max}]$. *Then, the tuple (A, afo) is a* state *of $G(f)$. For a global ordering π of $G(f)$, let $A^\pi : V(f) \to [|A|] \cup \mathsf{labset} f)$,*

$$A^\pi(v) \mapsto \begin{cases} i, & i \in [|A|],\ v = \mathsf{condense}(\pi, A)[i], \\ \mathsf{lab}(v), & else. \end{cases}$$

*A global ordering π of $G(f)$ is k-*activating *for (A, afo) if there is k-vertex set $S \subseteq V(G)$ such that for every vertex $v \in V(G) \setminus S$ we have that*

$$\pi_G^{<}(v) \ \geq \ \mathsf{thr}(v) - \mathsf{afo}(A^\pi(v)).$$

Example 3. Let $\mathsf{afo}(1) = 1$, and for $x \in \{2, \ldots, 6, \alpha, \beta, \gamma\}$, let $\mathsf{afo}(x) = 0$. The activation from outside for vertex v_1 is $\mathsf{afo}(A^\pi(v_1)) = \mathsf{afo}(1) = 1$ and for vertex v_{10} it is $\mathsf{afo}(A^\pi(v_{10})) = \mathsf{afo}(\alpha) = 0$. Further, π is 0-activating for state (A, afo).

We define nice orderings, analogously for global orderings π and local orderings A. As we show in the following, for every k-activating global ordering π there is a slightly modified k-activating global ordering π which is nice to every subexpression of f. Our local ordering A includes the $(t_{\max} + 1)$-st vertex of every label. Thus, whether π is to nice the current expression f is expressed in the ordering of A. Therefore, our algorithm can avoid not nice global orderings by ignoring states where the local ordering A is not nice to f.

Definition 5. *Let G be a graph with global ordering π. Let f be an ℓ-expression describing a subgraph of G. For label α, let $v_\alpha[1], v_\alpha[2], \dots \in V(f)$ be the vertices of label α of $G(f)$ ordered ascending according to π. For every label α, let $t^\alpha_{\max} := \min\{t_{\max}(G),\ |\{v \in V(G) \mid \mathsf{lab}(v) = \alpha\}|\}$. Then, π is* nice to f *if $f = \eta_{\alpha,\beta} f'$ implies that (if those respective positions exist)*

$$\pi(v_\alpha[t_{\max}{+}1]) \ > \ \pi(v_\beta[t^\beta_{\max}]) \quad and \quad \pi(v_\beta[t_{\max}{+}1]) \ > \ \pi(v_\alpha[t^\alpha_{\max}]).$$

Let A be the list of tuples of label and threshold of $\mathsf{condense}(\pi \restriction_{V(f)})$ *for graph $G(f)$, where $\pi \restriction_{V(f)}$ is π restricted to vertices $V(f)$. Then, A is nice to f if (and only if) π is nice to f.*

Example 4. Global ordering π is not nice to $\eta_{\beta,\gamma} f'$ since $\pi(v_\beta[t_{\max}+1]) = \pi(v_7) = 7 \not> 8 = \pi(v_8) = \pi(v_\gamma[t_{\max}])$. By switching the 7th and 8th position π becomes nice to $\eta_{\beta,\gamma} f'$. Likewise, A is not nice to $\eta_{\beta,\gamma} f'$, but $A' = \big((\beta,1), (\alpha,1), (\beta,1), (\alpha,2), (\alpha,2), (\gamma,2), \boldsymbol{(\gamma,2), (\beta,1)}, (\gamma,2)\big)$ is nice to $\eta_{\beta,\gamma} f'$.

Lemma 1. *Let f be an ℓ-expression and π a global ordering that is k-activating for graph $G(f)$. Then, there is a global ordering π' that is k-activating for graph $G(f)$ and nice to every subexpression of f.*

Proof (Sketch). There may be subexpressions $\eta_{\alpha,\beta} f'$ where the $(t_{\max}+1)$st vertex of label α is ordered before the first $t_{\max}$ vertices of label β, formally $\pi(v_\alpha[t_{\max}+1]) =: i < \pi(v_\beta[t^\beta_{\max}])$. We repair such a violation by moving all vertices of $v_\beta[1], \ldots, v_\beta[t^\beta_{\max}]$ that did not occur already between positions $(i-1)$ and i. Since there are $t_{\max}$ vertices of label α ordered before position i, the modified local ordering is still activating. We repair all such violations top-down. Following this order prevents recursive violations for already fixed subexpression $\eta_{\alpha',\beta'}$. For a full proof see online version.

Definition 6. *Graph $G(f)$ is k-*activating *for a state (A, afo) if there is a global ordering π that extends A, is k-activating for (A, afo), and is nice to every subexpression of f.*

Lemma 2. *Let f be an ℓ-expression. Then, graph $G(f)$ has a target set of size k if and only if there is a complete local ordering A of $G(f)$ such that $G(f)$ is k-activating for state $(A, \boldsymbol{0})$, where $\boldsymbol{0} \colon [|A|] \cup \mathsf{labels}(G) \to \{0\}$.*

Proof (Sketch). Use Lemma 1. For a full proof see online version.

It remains to specify the recursive dependency of our computation. We distinguish the three operations, which are adding edges if $f = \eta_{\alpha,\beta} f'$, taking the disjoint union if $f = f_1 \oplus f_2$, and re-labeling if $f = \rho_{\alpha\to\beta} f'$.

Consider a current ℓ-expression $\eta_{\alpha,\beta} f'$ and a state (A, afo). The operation $\eta_{\alpha,\beta}$ adds the edges between all vertices of label α and β. We adjust the activation from outside such that it replaces the edges between vertices of label α and β. The relative ordering of the first $(t_{\max}+1)$ vertices of label α and label β is already fixed by the local ordering A. We increase the activation from outside of a position y of A of label β for every prior position x of A of label α. For the activation from outside for vertex v of label α that is not referenced by A, every position x of A of label β increases the activation from outside. We denote the result as $\eta_{\alpha,\beta}\mathsf{afo}$.

Definition 7. *Let graph G with labels α and β have local ordering A. For $y \in [|A|] \cup \mathsf{labels}(G)$, let*

$$(\eta_{\alpha,\beta}\mathsf{afo})(y) := \min\big\{t_{\max},\ \mathsf{afo}(y) + \mathsf{add}(y)\big\}, \quad \textit{where}$$
$$\mathsf{add}(y) := \big|\big\{x \in [|A|] \ \big|\ x < y,\ \{\mathsf{lab}(x), \mathsf{lab}(y)\} = \{\alpha, \beta\}\big\}\big|,$$

where $1 < 2 < \cdots < |A| < \gamma$*, for every label* γ*; and where* $\mathsf{lab}(x)$*, for* $x \in [|A|]$*, is defined as* $\mathsf{lab}(A[x])$*. For every vertex* $v \in V(G)$*, let*

$$\mathsf{e}_\pi(v) := \left|\left\{u \in V(G) \;\middle|\; \pi(u) < \pi(v),\ \{\mathsf{lab}(u), \mathsf{lab}(v)\} = \{\alpha, \beta\}\right\}\right|.$$

The number of edges that additionally contribute to the activation of a vertex v, denoted by $\mathsf{e}_\pi(v)$, is equal to the increase of the activation from outside $\mathsf{add}(v)$ (while ignoring an overall activation exceeding $t_{\max}$).

Lemma 3. *Let global ordering* π *extend local ordering* A*, which is nice to* $\eta_{\alpha,\beta} f'$*. For every vertex* $v \in V(\eta_{\alpha,\beta} f')$*, we have that*

$$\min\{t_{\max},\ \mathsf{afo}(A^\pi(v)) + \mathsf{e}_\pi(v)\} = (\eta_{\alpha,\beta}\mathsf{afo})(A^\pi(v)).$$

Proof (Sketch). We need to show for every vertex v that the number of new neighbors ordered before, $\mathsf{e}_\pi(v)$, is equal to how much we increase $\mathsf{afo}(A^\pi(v))$, when capped by $t_{\max}$. Since A is nice to $\eta_{\alpha,\beta} f'$, this number of new neighbors is correctly expressed by comparing v with its neighbors of label β in A, which is how $\mathsf{add}(A^\pi(v))$ is computed. For a full proof see online version.

Lemma 4. *Graph* $G(\eta_{\alpha,\beta} f')$ *is* k*-activating for state* (A, afo) *if and only if* A *is nice to* $\eta_{\alpha,\beta} f'$ *and* $G(f')$ *is* k*-activating for* $(A, \eta_{\alpha,\beta}\mathsf{afo})$*.*

Proof (Sketch). We assume that the ℓ-expression $\eta_{\alpha,\beta} f'$ is irredundant as mentioned in the preliminaries. Then, every edge between vertices of label α and β is new to $G(f')$ such that $\pi^<_{\eta_{\alpha,\beta} f'}(v) = \pi^<_{f'}(v) + \mathsf{e}_\pi(v)$. For the forward direction, let $G(\eta_{\alpha,\beta} f)$ have global ordering π that extends A, is k-activating for state (A, afo) and nice to every subexpression of $\eta_{\alpha,\beta} f'$. It follows directly that A is nice to $\eta_{\alpha,\beta} A$. We in particular show that the same ordering π is k-activating for the modified state $(A, \eta_{\alpha,\beta}\mathsf{afo})$. That is, every non-target set vertex v has $\pi^<_{f'}(v) \geq \mathsf{thr}(v) - (\eta_{\alpha,\beta}\mathsf{afo})(A^\pi(v))$. We can follow this result from our initial observation and by applying Lemma 3. The backward direction is similar. For a full proof see online version.

In case of a current expression $f = f_1 \oplus f_2$, we have to show how to recursively rely on the subexpressions f_1 and f_2, analogously for $f = \rho_{\alpha\to\beta} f'$, on subexpression f'. For both cases, vertices of label β potentially come from different sets of vertices. In case of a re-labeling form α to β, a vertex of label β possibly had label α before or already had label β. In case of a disjoint union of subgraphs, a vertex of label β (or any other label) can be from either subgraph $G(f_1)$ or $G(f_2)$. For our indirect referenced vertices of our local ordering A, we do not know the true origin. Thus, we have to try all possible partitions of label β into labels α and β, respective all partitions of label β (and every other label) into either subgraph. As the possible local orderings A are bounded by our parameters, also the possible partitions are bounded by our parameters.

Definition 8

(1) A state (A, afo) *of graph* $G(f)$ completes *a state* (A^*, afo^*) *if* A *is complete, and removing from* A*, for every label* α*, the last tuples of label* α *from* A *until as many as in* A^* *remain, results in* A^**; and* $\mathsf{afo} : [|A|] \cup \mathsf{labels}(f) \to [0, t_{\max}]$*, maps* x *to* $\mathsf{afo}^*(x)$*, if defined for* x*, and otherwise to* $\mathsf{afo}^*\big(\mathsf{lab}(A[x])\big)$*.*

(2) Let $(f_1 \oplus f_2)$ *be an* ℓ*-expression. Then,* $\mathcal{S}[f_1 \oplus f_2, (A, \mathsf{afo})]$ *is the family of every pair of states* $\big((A_1, \mathsf{afo}_1), (A_2, \mathsf{afo}_2)\big)$ *that complete the possible incomplete states* $(A_1^*, \mathsf{afo}_1^*)$ *and* $(A_2^*, \mathsf{afo}_2^*)$ *that can be constructed as follows. Start with states* $(A_1^*, \mathsf{afo}_1^*)$*,* $(A_2^*, \mathsf{afo}_2^*)$ *where* $A_1^* = A_2^* = ()$ *and, for every label* α*, we have* $\mathsf{afo}_i^*(\alpha) = \mathsf{afo}(\alpha)$*. For position* j*, beginning from* 1 *to* $|A|$*, add* $A[j]$ *to the end of either list* $A_i^* \in \{A_1^*, A_2^*\}$ *where possible. For position* $j \in [|A|]$*, tuple* $A[j]$ *is added to list* A_i^**, and let* j' *be the position of* $A[j]$ *in* A_i^**. Then, let* $\mathsf{afo}_i^*(j') := \mathsf{afo}(j)$*.*

(3) Let (A, afo) *be a state of* $G(\rho_{\alpha\to\beta} f')$*. Then,* $\mathcal{S}[\rho_{\alpha\to\beta} f', (A, \mathsf{afo})]$ *is the family of every state* (A', afo') *that completes a state* (A^*, afo^*) *that can be constructed as follows. Re-label* $s \in [0, t^{\alpha}_{\max}(f')]$ *many tuples of* A *of label* β *to* α*, while at most* $t^{\beta}_{\max}(f')$ *of label* β *remain, resulting in* A^**. Let* afo^* *be defined as* afo *but where* $\mathsf{afo}^*(\alpha) = \mathsf{afo}(\beta)$*.*

Lemma 5. *Graph* $G(f_1 \oplus f_2)$ *is* k*-activating for state* (A, afo) *if and only if there are states* $\big((A_1, \mathsf{afo}_1), (A_2, \mathsf{afo}_2)\big) \in \mathcal{S}[f_1 \oplus f_2, (A, \mathsf{afo})]$ *and partition* $k_1 + k_2 = k$ *such that, for* $i \in \{1, 2\}$*, graph* $G(f_i)$ *is* k_i*-activating for* (A_i, afo_i)*.*

Lemma 6. *Graph* $G(\rho_{\alpha\to\beta} f')$ *is* k*-activating for state* (A, afo) *if and only if there is a state* $(A', \mathsf{afo}') \in \mathcal{S}[\rho_{\alpha\to\beta} f', (A, \mathsf{afo})]$ *such that* $G(f')$ *is* k*-activating for* (A', afo')*.*

Finally, we can show our main theorem, which was stated in the introduction.

Theorem 2 (Theorem 1 restated). *Let* $t_{\max}, \ell \in \mathbb{N}$*. There is an algorithm that, given a graph* G*, a threshold for each vertex* $\mathsf{thr} : V(G) \to [0, t_{\max}]$ *and an* ℓ*-expression* f *of* G*,computes the minimal size of a target set in time* $\mathcal{O}(\ell^{3\ell t} \cdot t^{\ell(4t+1)} \cdot |f|)$*, where* $t := t_{\max} + 1$ *and* $|f|$ *is the length of* f*.*

Proof. The minimal size of a target set is the minimal k of all local orderings A of $G(f)$ such that $G(f)$ is k-activating for $(A, \mathbf{0})$, as seen in Lemma 2.

Our algorithm computes the minimal k for possibly each subexpression f' of f and state (A, afo) of $G(f')$, in the fashion of dynamic programming. The minimum for a subexpression f' and state (A, afo) of $G(f')$ is remembered for future queries. There are at most $(\ell t)^{\ell t}$ possible local orderings A for a subgraph $G(f')$. And there are at most $t^{\ell t + \ell}$ possible activations from outside $\mathsf{afo} : [|A|] \cup \mathsf{labels}(f) \to [0, t_{\max}]$. Thus, there are at most $(\ell t)^{\ell t} \cdot t^{\ell t + \ell}$ different states for a fixed subexpression. Further, every computation is the minimum of at most $(\ell t)^{2\ell t}$ entries (an upper bound is guessing A_1, A_2 respectively A' from scratch), and the minimum can be found in linear time. Therefore, the algorithm runs in time $\mathcal{O}((\ell t)^{\ell t} \cdot t^{\ell t+\ell} \cdot (\ell t)^{2\ell t}) \cdot |f| = \mathcal{O}(\ell^{3\ell t} \cdot t^{\ell(4t+1)} \cdot |f|)$. If (A, afo) is not a correct state for $G(f')$, set its minimum to ∞.

If f contains only one operation, then $f = v(\alpha)$ and the only possible global ordering is $\pi : \{v\} \to \{1\}$. Graph $G(f)$ is at least 1-activating, and possibly 0-activating if $\mathsf{thr}(v) \geq \mathsf{thr}(v) - \mathsf{afo}(1)$. Answer accordingly in time $\mathcal{O}(1)$.

Otherwise, if f consists of more than one operation, we have either of the recursive cases that f is $\eta_{\alpha,\beta} f'$, $f_1 \oplus f_2$ or $\rho_{\alpha\to\beta} f'$. According to Lemmas 5 and 6 respectively, graph $G(f_1 \oplus f_2)$ is k-activating for state (A, afo) if and only if there is a pair of states $\big((A_1, \mathsf{afo}_1), (A_2, \mathsf{afo}_2)\big) \in \mathcal{S}[f_1 \oplus f_2, (A, \mathsf{afo})]$ and partition $k_1 + k_2 = k$ such that, for $i \in \{1, 2\}$, the graph $G(f_i)$ is k_i-activating for (A_i, afo_i);and graph $G(f\rho_{\alpha\to\beta} f')$ is k-activating if and only if there is a state $(A', \mathsf{afo}') \in \mathcal{S}[\rho_{\alpha\to\beta} f', (A, \mathsf{afo})]$ such that $G(f')$ is k-activating for (A', afo'). Therefore, in those two cases we can recursively obtain a minimum size of a target set by querying for the according subgraphs $G(f'), G(f_1), G(f_2)$ and states $\big((A_1, \mathsf{afo}_1), (A_2, \mathsf{afo}_2)\big) \in \mathcal{S}[f_1 \oplus f_2, (A, \mathsf{afo})]$ and $(A', \mathsf{afo}') \in \mathcal{S}[\rho_{\alpha\to\beta} f', (A, \mathsf{afo})]$, respectively. In case of $f = f_1 \oplus f_2$ the minimum size of a target set is the minimum of the sum of the minimum sizes for f_1 and f_2. For $f = \rho_{\alpha\to\beta} f$ the minimum size is equal to the minimum for f'.

According to Lemma 4, graph $G(\eta_{\alpha,\beta} f')$ is k-activating for state (A, afo) if and only if A is nice $\eta_{\alpha,\beta} f'$ and graph $G(\eta_{\alpha,\beta} f')$ is k-activating for state $(A, \eta_{\alpha,\beta}\mathsf{afo})$. Thus, in case of that A is not nice to f we can discard the current computation for a minimal size of a target set for the graph $G(\eta_{\alpha,\beta} f')$ and state (A, afo). Otherwise, the minimum size of a target set is equal to the minimum size of subgraph $G(f)$ with state $(A, \eta_{\alpha,\beta}\mathsf{afo})$.

4 Conclusion

In this work, we gave an FPT-algorithm for TSS for the combined parameters clique-width and maximum threshold. This result generalizes all previous FPT-membership results of TSS with constant thresholds. It would be interesting to explore the whole dichotomy of constant TSS for common structural parameters. Is there a different dichotomy when the maximum threshold is a parameter and not a constant?

References

1. Bazgan, C., Chopin, M., Nichterlein, A., Sikora, F.: Parameterized inapproximability of target set selection and generalizations. Computability **3**(2), 135–145 (2014)
2. Ben-Zwi, O., Hermelin, D., Lokshtanov, D., Newman, I.: Treewidth governs the complexity of target set selection. Discret. Optim. **8**(1), 87–96 (2011)
3. Betzler, N., Bredereck, R., Niedermeier, R., Uhlmann, J.: On bounded-degree vertex deletion parameterized by treewidth. Discret. Appl. Math. **160**(1–2), 53–60 (2012)
4. Chen, N.: On the approximability of influence in social networks. SIAM J. Discret. Math. **23**(3), 1400–1415 (2009)
5. Chopin, M., Nichterlein, A., Niedermeier, R., Weller, M.: Constant thresholds can make target set selection tractable. Theory Comput. Syst. **55**(1), 61–83 (2014)

6. Cicalese, F., Cordasco, G., Gargano, L., Milanic, M., Vaccaro, U.: Latency-bounded target set selection in social networks. Theor. Comput. Sci. **535**, 1–15 (2014)
7. Corneil, D.G., Rotics, U.: On the relationship between clique-width and treewidth. SIAM J. Comput. **34**(4), 825–847 (2005)
8. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discret. Appl. Math. **101**(1–3), 77–114 (2000)
9. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
10. Downey, R.G., Thilikos, D.M.: Confronting intractability via parameters. CoRR, abs/1106.3161 (2011)
11. Dvorák, P., Knop, D., Toufar, T.: Target set selection in dense graph classes. CoRR, abs/1610.07530 (2016)
12. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29953-X
13. Gajarský, J., Lampis, M., Ordyniak, S.: Parameterized algorithms for modular-width. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 163–176. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03898-8_15
14. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, D.C., USA, 24–27 August 2003, pp. 137–146 (2003)
15. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. Soc. Netw. Anal. Min. **3**(2), 233–256 (2013)
16. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2006)
17. Oum, S.: Approximating rank-width and clique-width quickly. ACM Trans. Algorithms **5**(1), 1–20 (2008)