Parameterized Approximability of Maximizing the Spread of Influence in Networks

Cristina Bazgan[1,3], Morgan Chopin[1], André Nichterlein[2], and Florian Sikora[1]

[1] PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243, France
{bazgan,chopin,florian.sikora}@lamsade.dauphine.fr
[2] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
andre.nichterlein@tu-berlin.de
[3] Institut Universitaire de France

Abstract. In this paper, we consider the problem of maximizing the spread of influence through a social network. Here, we are given a graph $G = (V, E)$, a positive integer k and a threshold value $\text{thr}(v)$ attached to each vertex $v \in V$. The objective is then to find a subset of k vertices to "activate" such that the number of activated vertices at the end of a propagation process is maximum. A vertex v gets activated if at least $\text{thr}(v)$ of its neighbors are. We show that this problem is strongly inapproximable in fpt-time with respect to (w.r.t.) parameter k even for very restrictive thresholds. For unanimity thresholds, we prove that the problem is inapproximable in polynomial time and the decision version is W[1]-hard w.r.t. parameter k. On the positive side, it becomes $r(n)$-approximable in fpt-time w.r.t. parameter k for any strictly increasing function r. Moreover, we give an fpt-time algorithm to solve the decision version for bounded degree graphs.

1 Introduction

Optimization problems that involve a diffusion process in a graph are well studied [18,14,8,1,12,7,3,19]. Such problems share the common property that, according to a specified *propagation rule*, a chosen subset of vertices activates all or a fixed fraction of the vertices, where initially all but the chosen vertices are inactive. Such optimization problems model the spread of influence or information in social networks via word-of-mouth recommendations, of diseases in populations, or of faults in distributed computing [18,14,12]. One representative problem that appears in this context is the *influence maximization* problem introduced by Kempe *et al.* [14]. Given a directed graph, the task is to choose a vertex subset of size at most a fixed number such that the number of activated vertices at the end of the propagation process is maximized. The authors show that the problem is polynomial-time $(\frac{e}{e-1} + \varepsilon)$-approximable for any $\varepsilon > 0$ under some stochastic propagation models, but NP-hard to approximate within a ratio of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ for general propagation rules.

In this paper, we use the following deterministic propagation model. We are given an undirected graph, a threshold value $\text{thr}(v)$ associated to each vertex

To appear in *Proceedings of the 9th Annual International Computing and Combinatorics Conference (COCOON'13)*, Hangzhou, China, June 2013.

v, and the following propagation rule: a vertex becomes active if at least $\mathrm{thr}(v)$ many neighbors of v are active. The propagation process proceeds in several rounds and stops when no further vertex becomes active. Given this model, finding and activating a minimum-size vertex subset such that all or a fixed fraction of the vertices become active is known as the *minimum target set selection* (MinTSS) problem introduced by Chen [8]. It has been shown NP-hard even for bipartite graphs of bounded degree when all thresholds are at most two [8]. Moreover, the problem was surprisingly shown to be hard to approximate within a ratio $O(2^{\log^{1-\varepsilon} n})$ for any $\varepsilon > 0$, even for constant degree graphs with thresholds at most two and for general graphs when the threshold of each vertex is half its degree (called *majority* thresholds) [8]. If the threshold of each vertex equals its degree (*unanimity* thresholds), then the problem is polynomial-time equivalent to the vertex cover problem [8] and, thus, admits a 2-approximation and is hard to approximate with a ratio better than 1.36 [10]. Concerning the parameterized complexity, the problem is shown to be W[2]-hard with respect to (w.r.t.) the solution size, even on bipartite graphs of diameter four with majority thresholds or thresholds at most two [16]. Furthermore, it is W[1]-hard w.r.t. each of the parameters "treewidth", "cluster vertex deletion number", and "pathwidth" [3,9]. On the positive side, the problem becomes fixed-parameter tractable w.r.t. each of the single parameters "vertex cover number", "feedback edge set size", and "bandwidth" [16,9]. If the input graph is complete, or has a bounded treewidth and bounded thresholds then the problem is polynomial-time solvable [16,3].

Here, we study the complementary problem of MinTSS, called *maximum k-influence* (Max$k$Inf) where the task is to maximize the number of activated vertices instead of minimizing the target set size. Since both optimization problems have the same decision version, the parameterized as well as NP-hardness results directly transfer from MinTSS to Max$k$Inf. We show that also Max$k$Inf is hard to approximate and, confronted with the computational hardness, we study the parameterized approximability of Max$k$Inf.

Our results. Concerning the approximability of the problem, there are two possibilities of measuring the value of a solution: counting the vertices activated by the propagation process including or excluding the initially chosen vertices (denoted by Max Closed k-Influence and Max Open k-Influence, respectively). Observe that whether or not counting the chosen vertices might change the approximation factor. In this paper, we consider both cases and our approximability results are summarized in Table 1.

While MinTSS is both constant-approximable in polynomial time and fixed-parameter tractable for the unanimity case, this does not hold anymore for our problem. Indeed, we prove that, in this case, Max Closed k-Influence (resp. Max Open k-Influence) is strongly inapproximable in polynomial-time and the decision version, denoted by (k,ℓ)-Influence, is W[1]-hard w.r.t. the combined parameter (k,ℓ). However, we show that Max Closed k-Influence (resp. Max Open k-Influence) becomes approximable if we are allowed to use fpt-time and (k,ℓ)-Influence gets fixed-parameter tractable w.r.t combined parameter (k,Δ), where Δ is the maximum degree of the input graph.

		Max Open k-Influence		Max Closed k-Influence	
Thresholds	Bounds	poly-time	fpt-time	poly-time	fpt-time
General	Upper	n	n	n	n
	Lower	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$
Constant	Upper	n	n	n	n
	Lower	$n^{\frac{1}{2}-\varepsilon}$,$\forall\varepsilon > 0$	$n^{\frac{1}{2}-\varepsilon}$,$\forall\varepsilon > 0$	$n^{\frac{1}{2}-\varepsilon}$,$\forall\varepsilon > 0$	$n^{\frac{1}{2}-\varepsilon}$,$\forall\varepsilon > 0$ [Th. 2]
Majority	Upper	n	n	n	n
	Lower	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$[Th. 1]
Unanimity	Upper	2^k [Th. 5]	$r(n), \forall r$ [Th. 2]	2^k	$r(n), \forall r$
	Lower	$n^{1-\varepsilon}$,$\forall\varepsilon > 0$ [Th. 4]	?	$1 + \varepsilon$ [Th. 7]	?

Table 1. Table of the approximation results for Max Open k-Influence and Max Closed k-Influence.

Our paper is organized as follows. In Section 2, after introducing some preliminaries, we establish some basic lemmas. In Section 3 we study Max Open k-Influence and Max Closed k-Influence with majority thresholds and thresholds at most two. In Section 4 we study the case of unanimity thresholds in general graphs and in bounded degree graphs. Conclusions are provided in Section 5. Due to space limitation, some proofs are deferred to a full version.

2 Preliminaries & Basic Observations

In this section, we provide basic backgrounds and notation used throughout this paper, give the statements of the studied problems, and establish some lemmas.

Graph terminology. Let $G = (V, E)$ be an *undirected graph.* For a subset $S \subseteq V$, $G[S]$ is the subgraph induced by S. The *open neighborhood* of a vertex $v \in V$, denoted by $N(v)$, is the set of all neighbors of v. The *closed neighborhood* of a vertex v, denoted $N[v]$, is the set $N(v) \cup \{v\}$. Furthermore, for a vertex set $V' \subset V$ we set $N(V') = \bigcup_{v \in V'} N(v)$ and $N[V'] = \bigcup_{v \in V'} N[v]$. The set $N^k[v]$, called the k-neighborhood of v, denotes the set of vertices which are at distance at most k from v (thus $N^1[v] = N[v]$). The *degree* of a vertex v is denoted by $\deg_G(v)$ and the *maximum degree* of the graph G is denoted by Δ_G. We skip the subscript if G is clear from the context. Two vertices are *twins* if they have the same neighborhood. They are called *true twins* if they are moreover neighbors, *false twins* otherwise.

Cardinality constrained problem. The problems studied in this paper are cardinality constrained. We use the notations and definitions from Cai [5]. A cardinality constrained optimization problem is a quadruple $A = (\mathcal{B}, \Phi, k, obj)$, where $\mathcal{B}$ is a finite set called solution base, $\Phi : 2^{\mathcal{B}} \to \{0, 1, 2, \ldots\} \cup \{-\infty, +\infty\}$ an objective function, k a non-negative integer and $obj \in \{min, max\}$. The goal is then to find a solution $S \subseteq \mathcal{B}$ of cardinality k so as to maximize (or minimize) the objective value $\Phi(S)$. If S is not a feasible solution we set $\Phi(S) = -\infty$ if $obj = max$ and $\Phi(S) = +\infty$ otherwise.

Parameterized complexity. A parameterized problem (I, k) is said *fixed-parameter tractable* (or in the class FPT) w.r.t. parameter k if it can be solved in $f(k) \cdot |I|^c$ time, where f is any computable function and c is a constant (one can see [11,17]). The parameterized complexity hierarchy is composed of the classes $\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \cdots \subseteq \mathsf{W[P]}$. A $\mathsf{W[1]}$-hard problem is not fixed-parameter tractable (unless $\mathsf{FPT} = \mathsf{W[1]}$) and one can prove $\mathsf{W[1]}$-hardness by means of a *parameterized reduction* from a $\mathsf{W[1]}$-hard problem. This is a mapping of an instance (I, k) of a problem A_1 in $g(k) \cdot |I|^{O(1)}$ time (for any computable g) into an instance (I', k') for A_2 such that $(I, k) \in A_1 \Leftrightarrow (I', k') \in A_2$ and $k' \leq h(k)$ for some h.

Approximation. Given an optimization problem Q and an instance I of this problem, we denote by $|I|$ the size of I, by $opt_Q(I)$ the optimum value of I and by $val(I, S)$ the value of a feasible solution S of I. The *performance ratio* of S (or *approximation factor*) is $r(I, S) = \max\left\{\frac{val(I,S)}{opt_Q(I)}, \frac{opt_Q(I)}{val(I,S)}\right\}$. The *error* of S, $\varepsilon(I, S)$, is defined by $\varepsilon(I, S) = r(I, S) - 1$. For a function f (resp. a constant $c > 1$), an algorithm is a $f(n)$-approximation (resp. a c-approximation) if for any instance I of Q it returns a solution S such that $r(I, S) \leq f(n)$ (resp. $r(I, S) \leq c$). An optimization problem is polynomial-time *constant approximable* (resp. has a *polynomial-time approximation scheme*) if, for some constant $c > 1$ (resp. every constant $\varepsilon > 0$), there exists a polynomial-time c-approximation (resp. $(1 + \varepsilon)$-approximation) for it. An optimization problem is *$f(n)$-approximable in fpt-time w.r.t. parameter k* if there exists an $f(n)$-approximation running in time $g(k) \cdot |I|^c$, where k is a positive integer depending on I, g is any computable function and c is a constant [15]. For a cardinality constrained problem a possible choice for the parameter is the cardinality of the solutions.

Problems definition. Let $G = (V, E)$ be an undirected graph and a threshold function $\text{thr} : V \to \mathbb{N}$. In this paper, we consider majority thresholds *i.e.* $\text{thr}(v) = \lceil \frac{\deg(v)}{2} \rceil$ for each $v \in V$, unanimity thresholds *i.e.* $\text{thr}(v) = \deg(v)$ for each $v \in V$, and constant thresholds *i.e.* $\text{thr}(v) \leq c$ for each $v \in V$ and some constant $c > 1$. Initially, all vertices are not activate and we select a subset $S \subseteq V$ of k vertices. The propagation unfolds in discrete steps. At time step 0, only the vertices in S are activated. At time step $t + 1$, a vertex v is activated if and only if the number of its activated neighbors at time t is at least $\text{thr}(v)$. We apply the rule iteratively until no more activations are possible. Given that S is the set of initially activated vertices, *closed activated vertices*, denoted by $\sigma[S]$ is the set of all activated vertices at the end of the propagation process and *closed activated vertices*, denoted by $\sigma(S)$, is the set $\sigma[S] \cup S$. The optimization problems we consider are then defined as follows.

MAX OPEN k-INFLUENCE **Input:** A graph $G = (V, E)$, a threshold function $\text{thr} : V \to \mathbb{N}$, and an integer k. **Output:** A subset $S \subseteq V$, $|S| \leq k$ such that $|\sigma(S)|$ is maximum.

Similarly, the Max Closed k-Influence problem asks for a set S such that $|\sigma[S]|$ is maximum. The corresponding decision version (k,ℓ)-Influence is also studied. Notice that, in this case, considering either the open or closed activated vertices is equivalent.

> (k,ℓ)-Influence
> **Input:** A graph $G=(V,E)$, a threshold function thr : $V \to \mathbb{N}$, and two integers k and ℓ.
> **Output:** Is there a subset $S \subseteq V$, $|S| \leq k$ such that $|\sigma(S)| \geq \ell$?

Basic results. In the following, we state and prove some lemmas that will be used later in the paper.

Lemma 1. *Let r be any computable function. If* Max Open k-Influence *is $r(n)$-approximable then* Max Closed k-Influence *is also $r(n)$-approximable where n is the input size.*

Proof. Let A be an $r(n)$-approximation algorithm for Max Open k-Influence. Let I be an instance of Max Closed k-Influence and $opt(I)$ its optimum value. When we apply A on I it returns a solution S such that $|\sigma(S)| \geq \frac{opt(I)-k}{r(n)}$ and then $|\sigma[S]| = k + |\sigma(S)| \geq \frac{opt(I)}{r(n)}$. □

Lemma 2. *Let I be an instance of a cardinality constrained optimization problem $A = (\mathcal{B}, \Phi, k, obj)$. If A is $r_1(k)$-approximable in fpt-time w.r.t. parameter k for some strictly increasing function r_1 then it is also $r_2(|\mathcal{B}|)$-approximable in fpt-time w.r.t. parameter k for any strictly increasing function r_2.*

Proof. Let r_1^{-1} and r_2^{-1} be the inverse functions of r_1 and r_2, respectively. We distinguish the following two cases.

Case 1: $k \leq r_1^{-1}(r_2(|\mathcal{B}|))$. In this case, we apply the $r_1(k)$-approximation algorithm and directly get the $r_2(|\mathcal{B}|)$-approximation in time $f(k) \cdot |\mathcal{B}|^{O(1)}$ for some computable function f.

Case 2: $k > r_1^{-1}(r_2(|\mathcal{B}|))$. We then have $|\mathcal{B}| < r_2^{-1}(r_1(k))$. In this case, we solve the problem exactly by brute-force. If $obj = \max$ (resp. $obj = \min$) then try all possible subset $S \subseteq \mathcal{B}$ of size k and take the one that maximizes (resp. minimizes) the objective value $\Phi(S)$. The running time is then $O(|\mathcal{B}|^k) = O(r_2^{-1}(r_1(k))^k)$.

The overall running time is $O(\max\{r_2^{-1}(r_1(k))^k, f(k) \cdot |\mathcal{B}|^{O(1)}\})$, that is, fpt-time. □

It is worth pointing out that a problem which is proven inapproximable in fpt-time obviously implies that it is not approximable in polynomial time with the same ratio. Therefore, fpt-time inapproximability can be considered as a "stronger" result than polynomial-time inapproximability.

3 Parameterized inapproximability

In this section, we consider the parameterized approximability of both Max Closed k-Influence and Max Open k-Influence. We show that these problems are W[2]-hard to approximate within $n^{1-\varepsilon}$ and $n^{\frac{1}{2}-\varepsilon}$ for any $\varepsilon \in (0,1)$ for majority thresholds and thresholds at most two, respectively. To do so, we use the following construction from Dominating Set as the starting point. The Dominating Set problem asks, given an undirected graph $G = (V, E)$ and an integer k, whether there is a vertex subset $S \subseteq V$, $|S| \leq k$, such that $N[S] = V$.

Basic Reduction. Given an instance $(G = (V, E), k)$ of Dominating Set we construct a bipartite graph $G' = (V', E')$ as follows. For each vertex $v \in V$ we add two vertices v^t and v^b (t and b respectively standing for *top* and *bottom*) to V'. For each edge $\{u, v\} \in E$ add the edge $\{v^t, w^b\}$. Finally, set $\text{thr}(v^t) = \deg_{G'}(v^t)$ and $\text{thr}(v^b) = 1$ for every top vertex v^t and every bottom vertex v^b, respectively. Clearly, the construction can be computed in polynomial time and, furthermore, it has the following property.

Lemma 3. *Let $G' = (V', E')$ be the graph obtained from a graph G using the above construction. Then G admits a dominating set of size k if and only if G' admits a subset $S' \subseteq V'$ of size k such that $\sigma[S'] = V'$.*

Inapproximability results. We are now ready to prove the main results of this section.

Theorem 1. *For any $\varepsilon \in (0,1)$, Max Closed k-Influence and Max Open k-Influence with majority thresholds cannot be approximated within $n^{1-\varepsilon}$ in fpt-time w.r.t. parameter k even on bipartite graphs, unless* FPT = W[2].

Proof. By Lemma 1, it suffices to show the result for Max Closed k-Influence. We construct a polynomial-time reduction from Dominating Set to Max Closed $(k+1)$-Influence with majority. In this reduction, we will make use of the ℓ-*edge* gadget, for some integer ℓ. An ℓ-edge between two vertices u and v consists of ℓ vertices of threshold one adjacent to both u and v.

Given an instance $I = (G = (V, E), k)$ of Dominating Set with $n = |V|$, $m = |E|$, we define an instance I' of Max Closed $(k+1)$-Influence. We start with the *basic reduction* and modify G' and the function thr as follows. Replace every edge $\{v^t, v^b\}$ by an $(k+2)$-edge between v^t and v^b. Moreover, for a given constant $\beta = \frac{8-5\varepsilon}{\varepsilon}$, let $L = \lceil n^{\beta} \rceil$ and we add nL more vertices $x_1^1, \ldots, x_n^1, \ldots, x_1^L, \ldots, x_n^L$. For $i = 1, \ldots, n$, vertex x_i^1 is adjacent to all the bottom vertices. Moreover, for any $j = 2, \ldots, L$, each x_i^j is adjacent to x_k^{j-1}, for any $i, k \in \{1, \ldots, n\}$. We also add a vertex w and an $n+(k+2)(\deg_G(v)-1)$-edge between w and v^b, for any bottom vertex v^b. For $i = 1, \ldots, n$, vertex x_i^1 is adjacent to w. For $i = 1, \ldots, n$ add n pending-vertices (*i.e.* degree one vertices) adjacent to x_i^L. For any vertex v^t add $(\deg_G(v)+1)(k+2)$ pending-vertices adjacent to

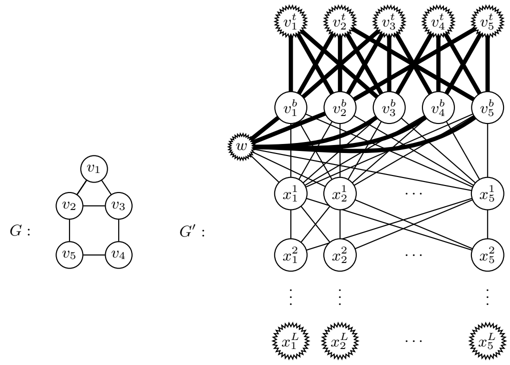

Fig. 1. The graph G' obtained after carrying out the modifications of Theorem 1. A thick edge represents an ℓ-edge for some $\ell > 0$. A "star" vertex v represents a vertex adjacent to $\frac{\deg_{G'}(v)}{2}$ pending-vertices.

v^t. Add also $n+n^2+(k+2)(2m-n)$ pending-vertices adjacent to w. All vertices of the graph G' have the majority thresholds (see also Figure 1).

We claim that if I is a *yes*-instance then $opt(I') \geq nL \geq n^{\beta+1}$; otherwise $opt(I') < n^4$. Let $n' = |V'|$, notice that we have $n' \leq n^4 + nL$.

Suppose that there exists a dominating set $S \subseteq V$ in G of size at most k. Consider the solution S' for I' containing the corresponding top vertices and vertex w. After the first round, all vertices belonging to the edge gadgets which top vertex is in S' are activated. Since S is a dominating set in G, after the second round, all the bottom vertices are activated. Indeed $\deg_{G'}(v^b) = 2(n + (k+2)\deg_G(v))$ and after the first round v^b has at least $k+2$ neighbors activated belonging to an $(k+2)$-edge between v^b and some $u^t \in V$ and $n+(k+2)(\deg_G(v)-1)$ neighbors activated belonging to an $n+(k+2)(\deg_G(v)-1)$-edge between v^b and w. Thus, every vertex x_i^1 gets active after the third round, and generally after the jth round, $j = 4, \ldots, L+2$ the vertices x_i^{j-2} are activated, and at the $(L+3)$th round all pending-vertices adjacent to x_i^L are activated. Therefore, the size of an optimal solution is at least $nL \geq n^{\beta+1}$.

Suppose that there is no dominating set in G of size k. Without loss of generality, we may assume that no pending-vertices are in a solution of I' since they all have threshold one. If w does not take part of a solution in I', then no vertex x_i^1 could be activated and in this case $opt(I')$ is less than $n' - nL \leq n^4$. Consider now the solutions of I' of size $k+1$ that contain w. Observe that if a top-vertex v^t gets active through bottom-vertices then v^t can not activate any other bottom-vertices. Indeed, as a contradiction, suppose that v^t is adjacent to a non-activated bottom-vertex. It follows that v^t could not have been activated because of its threshold and that no pending-vertices are part of the solution, a contradiction. Notice also that it is not possible to activate a bottom vertex by selecting some x_i^1 vertices since of their threshold. Moreover, since there is no dominating set of size k, any subset of k top vertices cannot activate all bottom vertices, therefore no vertex x_i^k, $i = 1, \ldots, n, k = 1, \ldots, L$ can be activated. Hence, less than $n' - nL$ vertices can be activated in G' and the size of an optimal solution is at most n^4.

Assume now that there is an fpt-time n^ε-approximation algorithm A for MAX CLOSED $(k+1)$-INFLUENCE with majority threshold. Thus, if I is a *yes*-instance, the algorithm gives a solution of value $A(I') \geq \frac{n^{\beta+1}}{(n')^{1-\varepsilon}} > \frac{n^{\beta+1}}{n^{(1-\varepsilon)(\beta+5)}} = n^4$ since $n' \leq n^4 + nL < n^5 L$. If I is a *no*-instance, the solution value is $A(I') < n^4$. Hence, the approximation algorithm A can distinguish in fpt-time between *yes*-instances and *no*-instances for DOMINATING SET implying that FPT = W[2] since this last problem is W[2]-hard [11]. □

Theorem 2. *For any $\varepsilon \in (0, \frac{1}{2})$,* MAX CLOSED k-INFLUENCE *and* MAX OPEN k-INFLUENCE *with thresholds at most two cannot be approximated within $n^{\frac{1}{2}-\varepsilon}$ in fpt-time w.r.t. parameter k even on bipartite graphs, unless* FPT = W[2].

Using Lemma 2, Theorem 1, and Theorem 2 we can deduce the following corollary.

Corollary 1. *For any strictly increasing function r,* MAX CLOSED k-INFLUENCE *and* MAX OPEN k-INFLUENCE *with thresholds at most two or majority thresholds cannot be approximated within $r(k)$ in fpt-time w.r.t. parameter k unless* FPT = W[2].

4 Unanimity thresholds

For the unanimity thresholds case, we will give some results on general graphs before focusing on bounded degree graphs and regular graphs.

4.1 General graphs

In this section, we first show that, in the unanimity case, (k, ℓ)-INFLUENCE is W[1]-hard w.r.t. parameter $k + \ell$ and MAX OPEN k-INFLUENCE is not approximable within $n^{1-\varepsilon}$ for any $\varepsilon \in (0, 1)$ in polynomial time, unless NP = ZPP. However, if we are allowed to use fpt-time then MAX OPEN k-INFLUENCE with unanimity is $r(n)$-approximable in fpt-time w.r.t. parameter k for any strictly increasing function r.

Theorem 3. (k, ℓ)-INFLUENCE *with unanimity thresholds is* W[1]*-hard w.r.t. the combined parameter (k, ℓ) even for bipartite graphs.*

Theorem 4. *For any $\varepsilon \in (0, 1)$,* MAX OPEN k-INFLUENCE *with unanimity thresholds cannot be approximated within $n^{1-\varepsilon}$ in polynomial time, unless* NP = ZPP.

Theorem 5. MAX OPEN k-INFLUENCE *and* MAX CLOSED k-INFLUENCE *with unanimity thresholds are 2^k-approximable in polynomial time.*

Using Lemma 2 and Theorem 5 we directly get the following.

Corollary 2. *For any strictly increasing function r,* Max Open k-Influence *and* Max Closed k-Influence *with unanimity thresholds are $r(n)$-approximable in fpt-time w.r.t. parameter k.*

For example, Max Open k-Influence is $\log(n)$-approximable in time $O^*(2^{k2^k})$.

Finding dense subgraphs. In the following we show that Max Open k-Influence with unanimity thresholds is at least as difficult to approximate as the Densest k-Subgraph problem, that consists of finding in a graph a subset of vertices of cardinality k that induces a maximum number of edges. In particular, any positive approximation result for Max Open k-Influence with unanimity would directly transfers to Densest k-Subgraph.

Theorem 6. *For any strictly increasing function r, if* Max Open k-Influence *with unanimity thresholds is $r(n)$-approximable in fpt-time w.r.t. parameter k then* Densest k-Subgraph *is $r(n)$-approximable in fpt-time w.r.t. parameter k.*

Using Theorem 6 and Corollary 2, we have the following corollary, independently established in [4].

Corollary 3. *For any strictly increasing function r,* Densest k-Subgraph *is $r(n)$-approximable in fpt-time w.r.t. parameter k.*

4.2 Bounded degree graphs and regular graphs

We show in the following that Max Open k-Influence and thus Max Closed k-Influence are constant approximable in polynomial time on bounded degree graphs with unanimity thresholds. Moreover, Max Closed k-Influence and then Max Open k-Influence have no polynomial-time approximation scheme even on 3-regular graphs if $\mathsf{P} \neq \mathsf{NP}$. Moreover, we show that (k,ℓ)-Influence is in FPT w.r.t. parameter k.

Lemma 4. Max Open k-Influence *and* Max Closed k-Influence *with unanimity thresholds on bounded degree graphs are constant approximable in polynomial time.*

Theorem 7. Max Open k-Influence *and* Max Closed k-Influence *with unanimity thresholds have no polynomial-time approximation scheme even on 3-regular graphs for $k = \theta(n)$, unless $\mathsf{P} = \mathsf{NP}$.*

In Theorem 3 we showed that (k,ℓ)-Influence with unanimity thresholds is W[1]-hard w.r.t. parameters k and ℓ. In the following we give several fixed-parameter tractability results for (k,ℓ)-Influence w.r.t. parameter k on regular graphs and bounded degree graphs with unanimity thresholds. First we show that using results of Cai *et al.* [6] we can obtain fixed-parameter tractable algorithms. Then we establish an explicit and more efficient combinatorial algorithm. Using [6] we can show:

Theorem 8. *(k, ℓ)-Influence with unanimity thresholds can be solved in $2^{O(k\Delta^3)}n^2 \log n$ time where Δ denotes the maximum degree and in $2^{O(k^2 \log k)}n \log n$ time for regular graphs.*

While the previous results use general frameworks to solve the problem, we now give a direct combinatorial algorithm for (k, ℓ)-Influence with unanimity thresholds on bounded degree graphs. For this algorithm we need the following definition and lemma.

Definition 1. *Let (α, β) be a pair of positive integers, $G = (V, E)$ an undirected graph with unanimity thresholds, and $v \in V$ a vertex. We call v a* realizing *vertex for the pair (α, β) if there exists a vertex subset $V' \subseteq N^{2\alpha-1}[v]$ of size $|V'| \leq \alpha$ such that $|\sigma(V')| \geq \beta$ and $\sigma[V']$ is connected. Furthermore, we call $\sigma[V']$ a realization of the pair (α, β).*

We show first that in bounded degree graphs the problem of deciding whether a vertex is a realizing vertex for a pair of positive integers (α, β) is fixed-parameter tractable w.r.t. parameter α.

Lemma 5. *Checking whether a vertex v is a realizing vertex for a pair of positive integers (α, β) can be done in $\Delta^{O(\alpha^2)}$ time, where Δ is the maximum degree.*

Consider in the following the Connected (k, ℓ)-Influence problem that is (k, ℓ)-Influence with the additional requirement that $G[\sigma[S]]$ has to be connected. Note that with Lemma 5 we can show that Connected (k, ℓ)-Influence is fixed parameter tractable w.r.t. parameter k on bounded degree graphs. Indeed, observe that two vertices in $\sigma(S)$ cannot be adjacent since we consider unanimity thresholds. From this and the requirement that $G[\sigma[S]]$ is connected, it follows that $G[\sigma[S]]$ has a diameter of at most $2k$. Hence, the algorithm for Connected (k, ℓ)-Influence checks for each vertex $v \in V$ whether v is a realizing vertex for the pair (k, ℓ). By Lemma 5 this gives an overall running time of $\Delta^{O(k^2)} \cdot n$.

We can extend the algorithm for the connected case to deal with the case where $G[\sigma[S]]$ is not connected. The general idea is as follows. For each connected component C_i of $G[\sigma[S]]$ the algorithm guesses the number of vertices in $S \cap C_i$ and in $\sigma(S) \cap C_i$. This gives an integer pair (k_i, ℓ_i) for each connected component in $G[\sigma[S]]$. Similar to the connected case, the algorithm will determine realizations for these pairs and the union of these realizations give S and $\sigma(S)$. Unlike the connected case, it is not enough to look for just one realization of a pair (k_i, ℓ_i) since the realizations of different pairs may be not disjoint and, thus, vertices may be counted twice as being activated. To avoid the double-counting we show that if there are "many" different realizations for a pair (k_i, ℓ_i), then there always exist a realization being disjoint to all realizations of the other pairs. Now consider only the integer pairs that do not have "many" different realizations. Since there are only "few" different realizations possible, the graph induced by all the vertices contained in all these realizations is "small". Thus, the algorithm can guess the realizations of the pairs having only "few" realizations and afterwards add greedily disjoint realizations of pairs having "many" realizations. See Algorithm 1 for the pseudocode.

Algorithm 1 The pseudocode of the algorithm solving the decision problem (k, ℓ)-Influence. The guessing part in the algorithm behind Lemma 5 is used in Line 7 as subroutine. The final check in Line 19 is done by brute force checking all possibilities.

1: **procedure** solveInfluence(G, thr, k, ℓ)
2:     Guess $x \in \{1, \ldots, k\}$ ▷ x: number of connected components of $G[\sigma[S]]$
3:     Guess $(k_1, \ell_1), \ldots, (k_x, \ell_x)$ such that $\sum_{i=1}^{x} k_i = k$ and $\sum_{i=1}^{x} \ell_i = \ell$
4:     Initialize $c_1 = c_2 = \ldots = c_x \leftarrow 0$ ▷ one counter for each integer pair (k_i, ℓ_i)
5:     **for** each vertex $v \in V$ **do** ▷ determine realizing vertices
6:         **for** $i \leftarrow 1$ **to** x **do**
7:             **if** v is a realizing vertex for the pair (k_i, ℓ_i) **then** ▷ see Lemma 5
8:                 $c_i \leftarrow c_i + 1$
9:                 $T(v, i) =$ "yes"
10:             **else**
11:                 $T(v, i) =$ "no"
12:     initialize $X \leftarrow \emptyset$ ▷ X stores all pairs with "few" realizations
13:     **for** $i \leftarrow 1$ **to** x **do**
14:         **if** $c_i \leq 2 \cdot x \cdot \Delta^{4k}$ **then**
15:             $X \leftarrow X \cup \{i\}$
16:     **for** each vertex $v \in V$ **do** ▷ remove vertices not realizing any pair in X
17:         **if** $\forall i \in X : T(v, i) =$ "no" **then**
18:             delete v from G.
19:     **if** all pairs (k_i, ℓ_i), $i \in X$, can be realized in the remaining graph **then**
20:         **return** 'YES'
21:     **else**
22:         **return** 'NO'

Theorem 9. *Algorithm 1 solves (k, ℓ)-Influence with unanimity thresholds in $2^{O(k^2 \log(k\Delta))} \cdot n$ time, where Δ is the maximum degree of the input graph.*

5 Conclusions

We established results concerning the parameterized complexity as well as the polynomial-time and fpt-time approximability of two problems modeling the spread of influence in social networks, namely Max Open k-Influence and Max Closed k-Influence.

In the case of unanimity thresholds, we show that Max Open k-Influence is at least as hard to approximate as Densest k-Subgraph, a well-studied problem. We established that Densest k-Subgraph is $r(n)$-approximable for any strictly increasing function r in fpt-time w.r.t. parameter k. An interesting open question consists of determining whether Max Open k-Influence is constant approximable in fpt-time. Such a positive result would improve the approximation in fpt-time for Densest k-Subgraph. In the case of thresholds bounded by two we excluded a polynomial time approximation scheme for Max

CLOSED k-INFLUENCE but we did not found any polynomial-time approximation algorithm. Hence, the question arises, whether this hardness result can be strengthened. Another interesting open question is to study the approximation of min target set selection problem in fpt-time.

References

1. A. Aazami and K. Stilp. Approximation algorithms and hardness for domination with propagation. *SIAM J Discrete Math*, 23(3):1382–1399, 2009.
2. P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theor Comput Sci*, 237(1-2):123–134, 2000.
3. O. Ben-Zwi, D. Hermelin, D. Lokshtanov, and I. Newman. Treewidth governs the complexity of target set selection. *Discrete Optim*, 8(1):87–96, 2011.
4. N. Bourgeois, A. Giannakos, G. Lucarelli, I. Milis, and V. T. Paschos. Exact and approximation algorithms for densest k-subgraph. In *Proc of WALCOM*, LNCS 7748, 2013. To appear.
5. L. Cai. Parameterized complexity of cardinality constrained optimization problems. *Comput J*, 51(1):102–121, 2008.
6. L. Cai, S. M. Chan, and S. O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proc of IWPEC*, LNCS 4169, pages 239–250, 2006.
7. C.-L. Chang and Y.-D. Lyuu. Spreading messages. *Theor Comput Sci*, 410(27–29):2714–2724, 2009.
8. N. Chen. On the approximability of influence in social networks. *SIAM J Discrete Math*, 23(3):1400–1415, 2009.
9. M. Chopin, A. Nichterlein, R. Niedermeier, and M. Weller. Constant thresholds can make target set selection tractable. In *Proc of MedAlg*, LNCS 7659, pages 120–133. Springer, 2012.
10. I. Dinur and S. Safra. The importance of being biased. In *Proc of STOC*, pages 33–42. ACM, 2002.
11. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
12. P. A. Dreyer and F. S. Roberts. Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Appl Math*, 157(7):1615 – 1627, 2009.
13. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math*, 182(1):105–142, 1999.
14. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc of KDD*, pages 137–146. ACM, 2003.
15. D. Marx. Parameterized complexity and approximation algorithms. *Comput J*, 51(1):60–78, 2008.
16. A. Nichterlein, R. Niedermeier, J. Uhlmann, and M. Weller. On tractable cases of target set selection. *Soc Network Anal Mining*, 2012. Online available.
17. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
18. D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theor Comput Sci*, 282:231–257, 2002.
19. T. V. T. Reddy and C. P. Rangan. Variants of spreading messages. *J Graph Algorithms Appl*, 15(5):683–699, 2011.

A Appendix

A.1 Proof 1 (Lemma 3)

Proof. The construction is illustrated on Figure 2. For the forward direction, suppose there exists a dominating set $S \subseteq V$ in G of size k. Consider the solution $S' \subseteq V'$ containing the corresponding top vertices. After the first round, all bottom vertices are activated since they have thresholds 1 and S is a dominating set. Finally, after the second round, all top vertices are activated too. For the reverse direction, suppose there is a subset $S' \subseteq V'$ of size k in G' such that $\sigma[S'] = V'$. We can assume w.l.o.g. that S' contains no bottom vertex. Since all bottom vertices are activated we have that $\{v_i : v_i^t \in S'\}$ is a dominating set in G. □

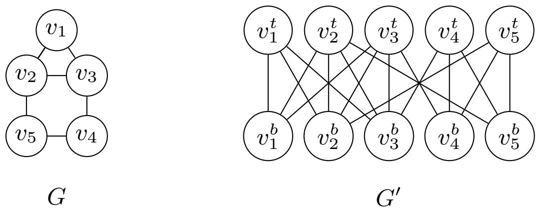

Fig. 2. Sample construction of the bipartite graph G' from a graph G of Dominating Set. All vertices v_i^t, $1 \le i \le 5$ have thresholds $\deg_{G'}(v_i^t)$ while all vertices v_i^b, $1 \le i \le 5$ have thresholds 1.

A.2 Proof 2 (Theorem 2)

Proof. By Lemma 1, it suffices to prove the result for Max Closed k-Influence. We construct a polynomial-time reduction from Dominating Set to Max Closed $(k+1)$-Influence with thresholds at most two. In this reduction, we will make use of the *directed edge* gadget. A directed edge from a vertex u to another vertex v consists of a 4-cycle $\{a, b, c, d\}$ such that a and u as well as c and v are adjacent. Moreover $\text{thr}(a) = \text{thr}(b) = \text{thr}(d) = 1$ and $\text{thr}(c) = 2$. The idea is that the vertices in the directed edge gadget become active if u is activated but not if v is activated. Hence, the activation process may go from u to v via the gadget but not in the reverse direction. In the rest of the proof, we may assume that no vertices from $\{a, b, c, d\}$ are part of a solution of Max Closed k-Influence. Indeed, it is always as good to take vertex u instead.

Given an instance $I = (G = (V, E), k)$ of Dominating Set with $n = |V|$, we define an instance I' of Max Closed k-Influence. We start with the *basic reduction* and modify G' and the function thr as follows. Set the thresholds of top-vertices to two. Replace every edge between a top vertex v^t and a bottom vertex v^b by a directed edge from v^t to v^b. For $j = 1, \ldots, n^\beta$ add a path

$p_1^j, \ldots, p_{n-1}^j$ of length $n-1$ and n^β *pending-vertices* with threshold one adjacent to p_{n-1}^j where $\beta = \frac{3.5-\varepsilon}{2\varepsilon}$. For $i = 2, \ldots, n$ insert a directed edge from v_i^b to all the vertices p_{i-1}^j, $j \in \{1, \ldots, n^\beta\}$. Furthermore, insert a directed edge from v_1^b to all the vertices p_1^j, $j \in \{1, \ldots, n^\beta\}$. Finally, set the thresholds of vertices lying in the added paths to two. This completes the construction (see Figure 3). Let $n' = |V'|$, notice that we have $n' < 2n + 4n^2 + 5n^{\beta+1} + n^\beta n^\beta < n^{\beta+2} + n^{2\beta}$

We claim that if I is a *yes*-instance then $opt(I') \geq n^{2\beta}$; otherwise $opt(I') < n^{\beta+3}$.

Suppose that there exists a dominating set $S \subseteq V$ in G of size at most k. Consider the solution S' for I' containing the corresponding top vertices. Since S is a dominating set in G, after the fourth round, all the bottom vertices are activated. Thus, after $n+6$ rounds, all the vertices in the paths are activated. It follows that in the next round all the pending-vertices are activated and the optimum value is then at least $n^{2\beta}$.

Suppose that there is no dominating set in G of size k. Consider a solution S' for I' of size k. Without loss of generality, we may assume that no pending-vertices or bottom vertices are contained in S' since they all have threshold one. For the reason previously mentioned, we know that no vertices from the directed edge gadgets are in S'. It follows that S' only contains top-vertices or vertices lying in the added paths. If the solution contains only top-vertices and since there is no dominating set of size k in G then at least one bottom-vertex is not activated. Moreover, because of the directed edges the activated bottom-vertices cannot activate new top-vertices. Thus at least one vertex of each path cannot be activated implying that no pending-vertices can be activated. This leads to a solution of size at most $n' - n^{2\beta} \leq n^{\beta+2}$. Now suppose that some vertices are taken from the added paths. Because of the directed edges, these last vertices cannot activate any bottom-vertices. Since at least one vertex of each path cannot be activated through its neighbors it follows that at most kn^β pending-vertices can be activated. The optimum solution is thus less than $n^{\beta+2} + kn^\beta < n^{\beta+3}$.

Assume now that there is an fpt-time $n^{\frac{1}{2}-\varepsilon}$-approximation algorithm A for Max Closed k-Influence with threshold at most two. Thus, if I is a *yes*-instance, the algorithm gives a solution of value $A(I') \geq \frac{n^{2\beta}}{n'^{0.5-\varepsilon}} > \frac{n^{2\beta}}{n^{(0.5-\varepsilon)(2\beta+1)}} > n^{\beta+3}$ since $n' \leq n^{\beta+2} + n^{2\beta} < n^{2\beta+1}$. If I is a *no*-instance, the solution value is $A(I') < n^{\beta+3}$. Hence, the approximation algorithm A can distinguish in fpt-time between *yes*-instances and *no*-instances for Dominating Set implying that $\mathsf{FPT} = \mathsf{W}[2]$ since this last problem is $\mathsf{W}[2]$-hard [11]. □

A.3 Proof 3 (Theorem 3)

Proof. We construct a fpt-reduction from Clique to (k, ℓ)-Influence. Given an instance $(G = (V, E), k)$ of Clique, we construct an instance $(G' = (V', E'), k, \ell)$ of (k, ℓ)-Influence as follows. For each vertex $v \in V$ add a copy v' to V'. For each edge $\{u, v\} \in E$, add $k+1$ edge-vertices $e_{uv}^1, \ldots, e_{uv}^{k+1}$ adjacent to both u' and v'. Set $\ell = (k+1)\binom{k}{2}$ and $\mathrm{thr}(u) = \deg_{G'}(u)$ for all $u \in V'$ (see also Figure 4).

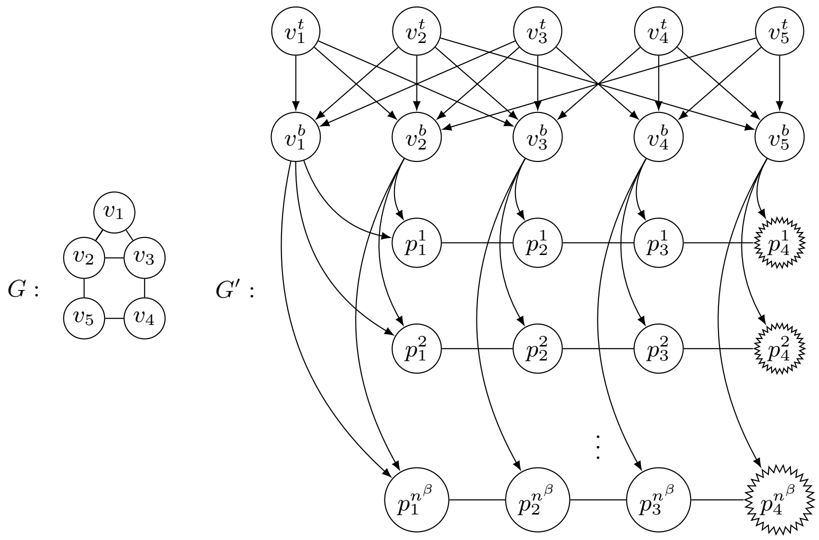

Fig. 3. The graph G' obtained from G after carrying out the modifications of Theorem 2. An arc (u, v) represents a directed edge gadget from u to v. A "star" vertex represents a vertex adjacent to n^β pending-vertices.

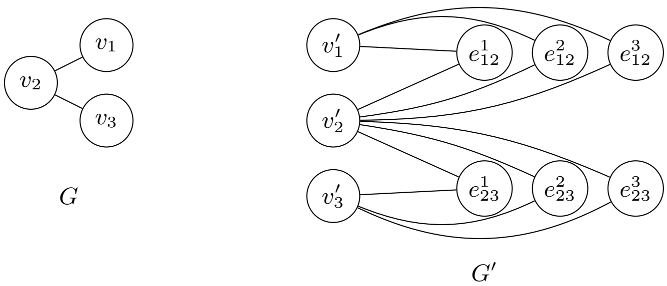

Fig. 4. Illustration of the reduction from an instance (G, k) of Clique to an instance (G', k, ℓ) of (k, ℓ)-Influence, where $k = 2$ and $\ell = 3$.

We claim that there is a clique of size k in G if and only if there exists a subset $S \subseteq V'$ of size k such that $|\sigma(S)| \geq \ell$.

"⇒": Assume that there is a clique $C \subseteq V$ of size k in G. One can easily verify that the set $S = \{v' \in V' : v \in C\}$ activates $|\sigma(S)| \geq (k+1)\binom{k}{2} = \ell$ edge-vertices in G' since C is clique.

"⇐": Suppose that there exists a subset $S \subseteq V'$ of size k such that $|\sigma(S)| \geq \ell$. We may assume without loss of generality that no edge-vertices belong to S. Indeed, each edge-vertex is adjacent to only vertices with threshold at least $k+1$. Thus choosing some edge-vertices to S cannot activate any new vertices in G'. Since the solution S activates at least $(k+1)\binom{k}{2}$ edge-vertices, this implies that S is a clique in G. □

A.4 Proof 4 (Theorem 4)

Proof. We will show how to transform any approximation algorithm for Max Open k-Influence into another one with the same ratio for Max Independent Set. Consider the instance I_k of Max Open k-Influence consisting of a graph $G = (V, E)$, an integer k and unanimity threshold. One can note and easily check that the following holds. Given a solution $S \subseteq V$ of I_k, $\sigma(S)$ is obtained in only one step of the diffusion process and is an independent set. Therefore there exists an integer $k^* \in [1, n]$ such that $\sigma(OPT(I_{k^*}))$ is the maximum independent set in G, where $OPT(I_{k^*})$ is the optimal solution of I_{k^*}.

Suppose that Max Open k-Influence has an $f(n)$-approximation algorithm A, we then have $|\sigma(A(I_{k^*}))| \geq \frac{|\sigma(OPT(I_{k^*}))|}{f(n)}$, where $A(I_{k^*})$ is a solution given by A for the instance I_{k^*}. It follows from the previous observation that $\sigma(A(I_{k^*}))$ is an independent set in G and an $f(n)$-approximate solution.

Now, it suffices to apply the approximation algorithm A for each $k = 1, \ldots, n$ and return the approximate solution $S_{\max}$ that has the largest value. Given this solution, we have $|\sigma(S_{\max})| \geq |\sigma(A(I_{k^*}))|$. Hence, we get a polynomial-time $f(n)$-approximation algorithm for Max Independent Set problem. Since Max Independent Set cannot be approximated within $n^{1-\varepsilon}$ for any $\varepsilon \in (0, 1)$ unless $\mathsf{NP} = \mathsf{ZPP}$ [13], the result follows. □

A.5 Proof 5 (Theorem 5)

Proof. By Lemma 1, it suffices to show the result for Max Open k-Influence. The polynomial-time algorithm consists in the following two steps: (i) Find F, the largest "false-twins set" such that $\deg(v) \leq k$, $\forall v \in F$, and (ii) Return $N(F)$. The first step can be done for example by searching for the largest set of identical lines with at most k ones in the adjacency matrix of the graph. Since F is a false-twins set with vertices of degree at most k, the size of the neighborhood of F is also bounded by k. Consider the activation of the set $N(F)$. After one round, this will activate $|\sigma(N(F))| \geq |F|$ vertices, since all the neighborhood of the vertices in F are activated.

To complete the proof, observe that for any target set of size at most k, there is at most 2^k different "false-twins sets". Therefore, any optimal solution could activate at most $2^k \cdot |F|$ vertices, providing the claimed approximation ratio. □

A.6 Proof 6 (Theorem 6)

The notion of an E-reduction (*error-preserving* reduction) was introduced by Khanna *et al.* 1999. A problem Π is called *E-reducible* to a problem Π', if there exist polynomial-time computable functions f, g and a constant β such that

- f maps an instance I of Π to an instance I' of Π' such that $opt(I)$ and $opt(I')$ are related by a polynomial factor, i.e. there exists a polynomial $p(n)$ such that $opt(I') \leq p(|I|)opt(I)$,
- g maps solutions S' of I' to solutions S of I such that $\varepsilon(I, S) \leq \beta\varepsilon(I', S')$.

An important property of an E-reduction is that it can be applied uniformly to all levels of approximability; that is, if Π is E-reducible to Π' and Π' belongs to $\mathcal{C}$ then Π belongs to $\mathcal{C}$ as well, where $\mathcal{C}$ is a class of optimization problems with any kind of approximation guarantee.

Proof. We give an E-reduction from Densest k-Subgraph to Max Open k-Influence. Consider an instance I of Densest k-Subgraph formed by a graph $G = (V, E)$ and we construct an instance I' of Max Open k-Influence with unanimity thresholds consisting of graph $G' = (V', E')$ as follows: for each vertex $v \in V$ add a copy v' to V'; for each edge $\{u, v\} \in E$ add a vertex e_{uv} to V', moreover add $k+1$ vertices $x_1, \ldots, x_{k+1}$. For any edge $\{u, v\} \in E$ add edges $\{u', e_{uv}\}, \{e_{uv}, v'\}$ to E', and add an edge between x_i and v' for any $1 \leq i \leq k+1$.

Let $S \subseteq V$, $|S| = k$ be an optimum solution for I that is $opt(I)$ is the number of edges induced by S. The set $S' = \{v' : v \in S\}$ is such that $\sigma(S) = opt(I)$ since no x vertex will be activated. Thus $opt(I') \geq opt(I)$.

Given any solution $S' \subseteq V'$ of size k, we can consider that S' contains only vertices of type v' such that $v \in V$. Thus the set $S = \{v : v' \in S'\}$ has value $val(S) = val(S')$. Moreover if S' is optimal, then $opt(I) \geq opt(I')$ and thus $opt(I) = opt(I')$. Therefore, we have $\varepsilon(I, S) = \varepsilon(I', S')$. □

A.7 Proof 7 (Lemma 4)

Proof. By Lemma 1, it suffices to show the result for Max Open k-Influence. Indeed on graphs of degree bounded by Δ, the optimum is bounded by $k \cdot \Delta$ and we can construct in polynomial time a solution S of value at least $\lfloor \frac{k}{\Delta} \rfloor$ by considering iteratively vertices with disjoint neighborhoods and putting their neighbors in S. □

A.8 Proof 8 (Theorem 7)

Proof. By Lemma 1, it suffices to show the result for Max Closed k-Influence. We show that if Max Closed k-Influence with unanimity thresholds has a polynomial-time approximation scheme $A_{\varepsilon'}$, $\varepsilon' \in (0,1)$, on 3-regular graphs when $k = \theta(n)$, then Min Vertex Cover has also a polynomial-time approximation scheme on 3-regular graphs. Consider $G = (V,E)$ a 3-regular graph. Clearly, a minimum vertex cover has a value $opt(G)$ satisfying $\frac{n}{2} \leq opt(G) < n$. For any $\varepsilon \in (0,1)$, we apply the polynomial-time approximation scheme $A_{\varepsilon'}$ that establishes an $(1+\varepsilon')$-approximation for Max Closed k-Influence on graph G for each k between $\frac{n}{2}$ and n and $\varepsilon' = \frac{\varepsilon}{2-\varepsilon}$. By applying $A_{\varepsilon'}$ on G for k between $\frac{n}{2}$ and n, we obtain a solution $S_k \subset V$ of size k such that $S_k \cup \sigma(S_k)$ is an $(1+\varepsilon')$-approximation. The set $V \setminus \sigma(S_k)$ is a vertex cover in G of size denoted by val_k. We show in the following that the best solution obtained in this way is an $(1+\varepsilon)$-approximation for Min Vertex Cover on G. Indeed the best solution obtained in this way has a value $val^* \leq val_\ell$, where val_ℓ is the value of the solution obtained for $\ell = opt(G)$. Thus $val_\ell = |V \setminus \sigma(S_\ell)|$. Since $|S_\ell \cup \sigma(S_\ell)|$ is an $(1+\varepsilon')$-approximation and the optimum solution activates all vertices, we have $|S_\ell \cup \sigma(S_\ell)| \geq \frac{n}{1+\varepsilon'}$ and $|V \setminus (S_\ell \cup \sigma(S_\ell))| \leq n\frac{\varepsilon'}{1+\varepsilon'}$. Thus $val^* \leq val_\ell \leq \ell + n\frac{\varepsilon'}{1+\varepsilon'} \leq \ell(1+\frac{2\varepsilon'}{1+\varepsilon'}) = \ell(1+\varepsilon)$. The theorem follows from the fact that Min Vertex Cover has no polynomial-time approximation scheme on 3-regular graphs, unless $\mathsf{P} = \mathsf{NP}$ [2]. ☐

A.9 Proof 9 (Theorem 8)

Proof. For graphs of maximum degree Δ, we simply apply the result from [6, Theorem 4] with $i = 3$.

Let G be a Δ-regular graph. When $\Delta > k$, any k vertices of the graph form a solution since no vertex outside the set becomes active. Hence, we assume in the following that $\Delta \leq k$. Since G is regular, it follows that any subset S, $|S| = k$ can activate at most k vertices. Hence, the graph $G[S \cup \sigma(S)]$ contains at most $2k$ vertices and, thus, $\ell \leq k$. Furthermore, since we consider unanimity thresholds, every vertex $v \in \sigma(S)$ has exactly Δ neighbors in S and, thus, $|N_{G[S\cup\sigma(S)]}(v)| = \Delta$ and $N_{G[S\cup\sigma(S)]}(v) \subseteq S$. Our fpt-algorithm solving (k,ℓ)-Influence runs in two phases:

Phase 1: Guess a graph H being isomorphic to $G[S \cup \sigma(S)]$.
Phase 2: Check whether H is a subgraph of G.

Phase 1 is realized by simply iterating over all possible graphs H with $k+\ell$ vertices. A simple upper bound on the number of different graphs with $k+\ell$ vertices is $2^{\binom{k+\ell}{2}} \leq 2^{4k^2}$. Hence, in Phase 1 the algorithm tries at most $O(2^{4k^2})$ possibilities. Note that Phase 2 can be done in $2^{O(\Delta k \log k)} n \log n$ using a result from [6, Theorem 1]. Altogether this gives a running time of $O(2^{4k^2} 2^{O(\Delta k \log k)} n \log n)$. Since $\Delta \leq k$, this gives $2^{O(k^2 \log k)} n \log n$. The correctness of the algorithm follows from the exhaustive search. ☐

A.10 Proof 10 (Lemma 5)

Proof. The algorithm solving the problem checks for all vertex subsets V' of size α in $N^{2\alpha-1}[v]$ whether V' activates at least β vertices and whether $\sigma[V']$ is connected. Since we consider unanimity thresholds it follows that $\sigma[V'] \subseteq N^{2\alpha}[v]$.

The correctness of this algorithm results from the exhaustive search. We study in the following the running time: The $(2\alpha-1)^{\text{th}}$ neighborhood of any vertex contains at most $\Delta(\Delta^{2\alpha})/(\Delta-1)+1 \leq 2\Delta^{2\alpha}$ vertices. Hence, there are $2^{\alpha}\Delta^{(2\alpha)\alpha}$ possibilities to choose the α vertices forming V'. For each choice of V' the algorithm has to check how many vertices are activated by V'. Since this can be done in linear time and there are $O(\Delta\Delta^{2\alpha})$ edges, this gives another $O(\Delta^{2\alpha+1})$ term. Altogether, we obtain a running time of $O(2^{\alpha}\Delta^{2\alpha^2+2\alpha+1})$ $= \Delta^{O(\alpha^2)}$. □

A.11 Proof 11 (Theorem 9)

Proof. Let S be a solution set, that is, $S \subset V$, $|S| \leq k$ and $\sigma(S) \geq \ell$. In the following we show that Algorithm 1 decides whether such set S exists or not in $2^{O(k^2 \log(k\Delta))} \cdot n$ time. We remark that the algorithm can be adapted to also give such set S if it exists. First we prove the correctness of the algorithm and then show the running time bound.

Correctness: We now show that a solution set S exists if and only if the algorithm returns "YES".

"$\Rightarrow$:" Assume that S is the solution set. Observe that $G[\sigma[S]]$ consists of at most k connected components and, thus, the guesses in Lines 2 and 3 are correct. Clearly, in the solution set S there is a realization for each pair (k_i, ℓ_i). Furthermore observe that in Line 13 it holds that $X \subseteq \{1, \ldots, x\}$ and that in the loop starting in Line 16 only vertices that cannot realize any pair corresponding to X are deleted. Hence, there exists a realization for the pairs corresponding to X in the remaining graph. Since the checking in Line 19 is done by trying all possibilities, the algorithm returns "YES".

"$\Leftarrow$:" Now assume that the algorithm returns "YES". Observe that this implies that in Line 19 there exists a realization for the all the pairs corresponding to X. Hence, it remains to show that for each pair $j \in \{1, \ldots, x\} \setminus X$ there exists a realization in G. (Clearly, if all pairs are realized then the union of the realizations form the vertex set $\sigma[S]$ such that $|S| = k$.) To see that there exist realizations for these pairs observe the following: The $(4k)^{\text{th}}$ neighborhood of any vertex contains at most $2\Delta^{4k}$ vertices. Thus, if in the case of two pairs $(k_1, \ell_1), (k_2, \ell_2)$ the value of the second counter is $c_2 > 2\Delta^{4k}$, then we can deduce that for every realizing vertex v_1 for (k_1, ℓ_1) there exists a realizing vertex v_2 for (k_2, ℓ_2) such that the distance d between v_1 and v_2 is more than $4k$. Since $d > 4k$, it follows that the realizations for (k_1, ℓ_1) and (k_2, ℓ_2) do not overlap. (If two realizations would overlap then some vertices in $\sigma(S)$ may be counted twice.) Generalizing this argument to x integer pairs $(k_1, \ell_1), \ldots, (k_x, \ell_x)$ yields the following: If there exists an $i \in \{1, \ldots, x\}$ such that $c_i > x \cdot 2 \cdot \Delta^{4k}$, then for any realization of the pairs (k_j, ℓ_j) with $i \neq j$ there exists a non-overlapping realization of (k_i, ℓ_i). Thus, we can ignore the pair (k_i, ℓ_i) where $c_i > x \cdot 2 \cdot \Delta^{4k}$ in the remaining algorithm and can assume that (k_i, ℓ_i) is realized.

Observe that from the Lines 5 to 16 it follows that for all $j \in \{1, \ldots, x\} \setminus X$ we have $c_j > x \cdot 2 \cdot \Delta^{4k}$. Thus, from the argumentation in the previous paragraph it follows that there exist non-overlapping realizations for all pairs corresponding to $\{1, \ldots, x\} \setminus X$. Thus, there exists a solution set S as required.

Running time: Observe that $\ell \leq \Delta k$ as described in the proof of Lemma 4. Thus, the guessing in Lines 2 and 3 can clearly be done in $O(k \cdot k^k (\Delta k)^k) = O(k^{2k+1} \Delta^k)$. By Lemma 5 the checking in Line 7 can be done in $\Delta^{O(k_i^2)}$ time. Thus, the loop in Line 5 requires $n \cdot \sum_{i=1}^{x} \Delta^{O(k_i^2)} \leq \Delta^{O(k^2)} \cdot x \cdot n$ time. Clearly, the loop in Line 13 needs $O(x) \leq O(k)$ time. Furthermore, the loop in Line 16 needs $O(k \cdot n)$ time. For the checking in Line 19 observe the following. After deleting the vertices in the loop in Line 16 the remaining graph can have at most $\sum_{i \in X} c_i \leq x \cdot 2 \cdot x \cdot \Delta^{4k}$ vertices. Furthermore, $\sum_{i \in X} k_i \leq k$ and, thus, there are at most $(2 \cdot x^2 \cdot \Delta^{4k})^k$ candidate subsets for the solution set S. Checking whether $\sum_{i \in X} k_i$ chosen vertices activate $\sum_{i \in X} \ell_i$ other vertices can be done in $(2 \cdot x^2 \cdot \Delta^{4k})^2$ time. Hence, the checking in Line 19 can be done in $\Delta^{O(k^2)}$ time. Putting all together we arrive at a running time of $(k\Delta)^{O(k^2)} \cdot n = 2^{O(k^2 \log(k\Delta))} \cdot n$. □