The parameterized complexity of *k*-edge induced subgraphs ⋆

Bingkai Lin[1] and Yijia Chen[1]

Shanghai Jiao Tong University
bing314159@sjtu.edu.cn, yijia.chen@cs.sjtu.edu.cn

Abstract. We prove that finding a k-edge induced subgraph is fixed-parameter tractable, thereby answering an open problem of Leizhen Cai [2]. Our algorithm is based on several combinatorial observations, Gauss' famous *Eureka* theorem [1], and a generalization of the well-known fpt-algorithm for the model-checking problem for first-order logic on graphs with locally bounded tree-width due to Frick and Grohe [13]. On the other hand, we show that two natural counting versions of the problem are hard. Hence, the k-edge induced subgraph problem is one of the very few known examples in parameterized complexity that are easy for decision while hard for counting.

1 Introduction

Induced subgraphs are one of the most natural substructures in graphs. They capture many different combinatorial objects, e.g., clique, independent set, chordless path. Thus, a great number of algorithmic problems are about finding certain induced subgraphs, and their complexity is among the mostly extensively studied in algorithmic graph theory [3, 7, 14]. In this paper, we are mainly interested in the problem of finding an induced subgraph which contains exactly k edges, i.e., a k-edge induced subgraph. This problem is equivalent to solving a special 0-1 quadratic Diophantine equation $x^T Ax = k$, where A is the adjacent matrix of G, $x \in \{0, 1\}^n, n = |V(G)|$.

It is not difficult to prove that the k-edge induced subgraph problem is NP-hard by a reduction from the clique problem. So we approach the problem via parameterized complexity [9, 12] and treat k as the parameter:

> p-Edge-Induced-Subgraph
> *Instance:* A graph G and $k \in \mathbb{N}$.
> *Parameter:* k.
> *Problem:* Decide whether G contains a k-edge induced subgraph.

As the main result of our paper, we show that p-Edge-Induced-Subgraph is fixed-parameter tractable. In fact, there are special cases of p-Edge-Induced-Subgraph whose fixed-parameter tractability has been known for a while. Since

⋆ Full version available at http://arxiv.org/abs/1105.0477 we can define a k-edge induced subgraph by a first-order sentence, using logic machinery, it can be shown that p-Edge-Induced-Subgraph is fixed-parameter tractable if the graph G has bounded tree-width [8], bounded local tree-width [13], etc., or most generally locally bounded expansion [10]. Unfortunately, the class of all graphs containing a k-edge induced subgraph does not possess any of these bounded measures. As another previously known case, using his *Random Separation* method [5] and Ramsey's Theorem, Cai [4] gave a very nice combinatorial algorithm that solves p-Edge-Induced-Subgraph when the parameter k is a *triangular number*, i.e., $k = \binom{m}{2}$ for some $m \in \mathbb{N}$. However, it looks very difficult to adapt Cai's algorithm to handle arbitrary k. Therefore neither logic nor combinatorial approach so far seems to be sufficient to settle the complexity of p-Edge-Induced-Subgraph by its own. So our fpt-algorithm is a rather tricky combination of these two methods.

1.1 Our approach

As just mentioned, our starting pointing is that the existence of a k-edge induced subgraph can be characterized by a sentence of first-order logic (FO) which depends on k only. It is a well-known result of Frick and Grohe [13] that the model-checking problem for FO on graphs of bounded *local tree-width* is fixed-parameter tractable. The local tree-width for a graph is a function bounding the tree-width of the induced subgraphs on the neighborhoods within a certain radius of every vertex. For instance, bounded-degree graphs have bounded local tree-width. These give immediately the fixed-parameter tractability of p-Edge-Induced-Subgraph on graphs with bounded degree[1].

With some more efforts, the above result can be extended to graphs G with degree bounded by a function of the parameter k. In that case, we can say the degree $\deg(v)$ of each vertex v is sufficiently small. The corresponding fpt-algorithm generalizes Frick and Grohe's Theorem to graphs with local tree-width bounded by a function of both the radius of the neighborhoods and an additional parameter. As a dual, if $\deg(v)$ of each vertex v in G is sufficiently large, or more precisely, the complement of G has degree bounded by a function of k, then we can decide p-Edge-Induced-Subgraph in fpt time, too.

Moving one step further, we consider graphs in which each $\deg(v)$ is either sufficiently small or sufficiently large, e.g., an n-star. We call such graphs *degree-extreme*. Using the same logic machinery as above, we are able to show the fixed-parameter tractability of p-Edge-Induced-Subgraph on degree-extreme graphs.

Assume that the graph G is not degree-extreme, i.e., there exists a vertex v_0 whose degree is neither sufficiently small nor sufficiently large. We partition the vertex set of G into two sets V_1 and V_2, where V_1 contains all vertices adjacent to v_0 and V_2 the remaining vertices. Then both V_1 and V_2 are relatively large. Note

[1] This is also a direct consequence of Seese's result that the model-checking problem for FO on bounded-degree graphs is fixed-parameter tractable [15]. But we find it more natural to work with bounded local tree-with in the following generalization.

possibly there are many edges between V_1 and V_2. Nevertheless, we can compute a vertex set B in G such that every edge between V_1 and V_2 has one vertex in B; and if B is large enough, we can show that G contains a k-edge induced subgraph. Otherwise, the graph G consists of two induced subgraphs $G[V_1]$ and $G[V_2]$, plus the edges between V_1 and V_2 adjacent to the set B of bounded size. In case $G[V_1]$ and $G[V_2]$ are both degree-extreme, we call such a graph G a *bridge* (of two degree-extreme graphs). By the logic method again, we prove that p-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable on bridges.

Now we are left with the case that at least one of $G[V_1]$ and $G[V_2]$ is not degree-extreme, say $G[V_1]$. Then we repeat the above procedure on $G[V_1]$ to get a partition $V_{11} \mid V_{12}$ of V_1. And again, both V_{11} and V_{12} are sufficiently large. Arguing as before, either we already know $G[V_1]$, and hence G, contains a k-edge induced subgraph, or there is a set B_1 of bounded size such that every edge between V_{11} and V_{22} intersects B_1.

Finally we remove the vertex set $B_0 := B \cup B_1$ from G. Then $G[V \setminus B_0]$ is the disjoint union of $G[V_{11} \setminus B_0]$, $G[V_{12} \setminus B_0]$ and $G[V_2 \setminus B_0]$. Moreover, all the three induced subgraphs are so large that, by Ramsey's Theorem, either one of them contains a large independent set, or we have three large disjoint cliques which are not adjacent to each other. For both cases, we show that $G[V \setminus B_0]$, and hence G, contains a k-edge induced subgraph. As a matter of fact, the second case is an easy consequence of a famous number-theoretic result of Gauss which states that *every natural number is the sum of three triangular numbers.*

We should mention that the running time of our algorithm in terms of the parameter k is *triple exponential* at least. On the other hand, it is linear in the size of the graph. We leave the detailed analysis in the full version of the paper.

1.2 Counting k-edge induced subgraphs

We also study the parameterized complexity of computing the number of k-edge induced subgraphs. For most natural problems, if the decision version is easy, then so is the counting problem. However, it turns out that two natural counting versions of p-EDGE-INDUCED-SUBGRAPH are both hard. To the best of our knowledge, there are only few natural problems which exhibit such a phenomenon [11, 6].

1.3 Organization of our paper

In Section 2 we introduce necessary background and fix our notations. We prove all required combinatorial results in Section 3. In particular, we present several simple structures in a graph which, if exist, guarantee the existence of a k-edge induced subgraph. Then in Section 4 we establish the fixed-parameter tractability of p-EDGE-INDUCED-SUBGRAPH on the degree-extreme graphs and the bridges using model-checking problems for FO. We present our fpt-algorithm for p-EDGE-INDUCED-SUBGRAPH by putting all the pieces together in Section 5. Finally in Section 6 we prove the hardness of the counting problems. Due to the space limitations, for some proofs we refer to the full version of this paper.

2 Preliminaries

$\mathbb{N}$ and $\mathbb{N}^+$ denote the sets of natural numbers (that is, nonnegative integers) and positive integers, respectively. For a natural number n let $[n] := \{1, \ldots, n\}$. We denote the alphabet $\{0, 1\}$ by Σ and identify problems with subsets Q of Σ^*. Clearly, as done mostly, we present concrete problems in a verbal, hence uncodified form over Σ. For every set S we use $|S|$ to denote its size. Moreover we let $\binom{S}{2}$ be the set of all two-element subsets of S, i.e., $\big\{\{a, b\} \bigm| a, b \in S \text{ and } a \neq b\big\}$. A triangular number is $\binom{k}{2} := \big|\binom{[k]}{2}\big|$ for some $k \in \mathbb{N}$. In particular, $\binom{0}{2} = \binom{1}{2} = 0$.

2.1 Parameterized complexity

A *parameterized problem* is a pair (Q, κ) consisting of a classical problem $Q \subseteq \Sigma^*$ and a polynomial time computable *parameterization* $\kappa : \Sigma^* \to \mathbb{N}$.

An algorithm $\mathbb{A}$ is an fpt-*algorithm with respect to a parameterization* κ if for every $x \in \Sigma^*$ the running time of $\mathbb{A}$ on x is bounded by $f(\kappa(x)) \cdot |x|^{O(1)}$ for a computable function $f : \mathbb{N} \to \mathbb{N}$. Or equivalently, we say that the algorithm $\mathbb{A}$ runs in fpt time. A parameterized problem (Q, κ) is *fixed-parameter tractable* if there is an fpt-algorithm with respect to κ that decides Q.

2.2 Graphs

We only consider *simple* graphs, that is, finite nonempty undirected graphs without loops and parallel edges. Every graph $G = (V, E)$ is thus determined by a nonempty vertex set V and an edge set $E \subseteq \binom{V}{2}$. For an edge $\{u, v\} \in E$ we say that u is *adjacent* to v, and vice versa. Often we also use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of G, respectively.

Let $G = (V, E)$ be a graph. For every vertex $v \in V$ the set $N^G(v)$ contains all vertices in G that are adjacent to v, i.e., $N^G(v) := \big\{u \bigm| \{u, v\} \in E\big\}$. Moreover, for every $S \subseteq V$ we let $N^G(S) := \bigcup_{v \in S} N^G(v)$. Note the degree of v, written $\deg^G(v)$, is $|N^G(v)|$. If $\deg^G(v) = 0$, then v is an *isolated* vertex. The distance $d^G(u, v)$ between two vertices $u, v \in V$ is the length of a shortest path from u to v in the graph G. If it is clear from the context, we omit the superscript G in the above notations and write $N(v)$, $\deg(v)$, etc., instead.

Every nonempty subset $S \subseteq V(G)$ induces a subgraph $G[S]$ with the vertex set S and the edge set $E(G[S]) := \binom{S}{2} \cap E(G)$. Consequently, a graph H is an *induced subgraph of* G if $H = G[V(H)]$. Recall that H is a k-*edge* induced subgraph of G for $k := |E(H)|$.

Again, let S be a set of vertices in G. Then S is a *clique*, if for every $u, v \in S$ we have either $u = v$ or $\{u, v\} \in E(G)$. On the other hand, the set S is an *independent set* in G, if $\{u, v\} \notin E(G)$ for all $u, v \in S$. For every $k \in \mathbb{N}$, there exists a constant $\mathscr{R}_k$, known as the *Ramsey number*, such that every graph G with $|V(G)| \geq \mathscr{R}_k$ has either a clique of size k or an independent set of size k. It is well-known that $\mathscr{R}_k < 2^{2 \cdot k}$ for every $k \in \mathbb{N}$.

2.3 Relational structures and first-order logic

A *vocabulary* τ is a finite set of relation symbols. Each relation symbol has an *arity*. A *structure* $\mathcal{A}$ of vocabulary τ, or simply structure, consists of a nonempty set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^r$ of each r-ary relation symbol $R \in \tau$. For example, a graph G can be identified with a structure $\mathcal{A}(G)$ of vocabulary $\tau_{\text{graph}} := \{E\}$ with the binary relation symbol E such that $A(G) := V(G)$ and $E^{\mathcal{A}(G)} := \{(u,v) \mid \{u,v\} \in E(G)\}$.

The *disjoint union* of two τ-structures $\mathcal{A}_1$ and $\mathcal{A}_2$ is again a τ-structure, denoted by $\mathcal{A}_1 \dot\cup \mathcal{A}_2$, whose universe is $A_1 \dot\cup A_2$, and where for each relation symbol $R \in \tau$ we let $R^{\mathcal{A}_1 \dot\cup \mathcal{A}_2} := R^{\mathcal{A}_1} \dot\cup R^{\mathcal{A}_2}$.

Let $\mathcal{A}$ be a structure of a vocabulary τ. Then the *Gaifman graph* of $\mathcal{A}$ is $G(\mathcal{A}) := (V, E)$ with $V := A$ and $E := \{\{a,b\} \mid a, b \in A, a \neq b, \text{and for some} R \in \tau, \text{and some tuple}(a_1, \ldots, a_r) \in R^{\mathcal{A}}, \{a,b\} \subseteq \{a_1, \ldots, a_r\}\}$. Note any unary relation in $\mathcal{A}$ has no influence on E.

Let $r \in \mathbb{N}$ and $a \in A$. Then the *r-neighborhood* of a is $N_r^{\mathcal{A}}(a) := \{b \in A \mid d^{G(\mathcal{A})}(a,b) \leq r\}$. Moreover, the structure $\mathcal{N}_r^{\mathcal{A}}(a)$ induced by the r-neighborhood of a has universe $N_r^{\mathcal{A}}(a)$, and for each r-ary relation symbol $R \in \tau$ the interpretation $\{(a_1, \ldots, a_r) \in R^{\mathcal{A}} \mid a_1, \ldots, a_r \in N_r^{\mathcal{A}}(a)\}$.

Formulas of first-order logic of vocabulary τ are built up from atomic formulas $x = y$ and $Rx_1 \ldots x_r$ where $x, y, x_1, \ldots, x_r$ are variables and $R \in \tau$ is of arity r, using the boolean connectives and existential and universal quantification.

2.4 Tree-width and local tree-width

We assume that the reader is familiar with the notion of *tree-width* $\text{tw}(G)$ of a graph G. Recall that the tree-width $\text{tw}(\mathcal{A})$ of a structure $\mathcal{A}$ is simply $\text{tw}(G(\mathcal{A}))$, that is, the tree-width of the Gaifman graph of $\mathcal{A}$. In fact, to understand most parts of our proofs and algorithms, it is sufficient to know that for every structure $\mathcal{A}$ we have $\text{tw}(\mathcal{A}) < |A|$.

Now we are ready to define the *local tree-width* of a structure $\mathcal{A}$. For every $r \in \mathbb{N}$ let $\text{ltw}(\mathcal{A}, r) := \max\{\text{tw}(\mathcal{N}_r^{\mathcal{A}}(a)) \mid a \in A\}$. Let $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a function and $p \in \mathbb{N}$. We say a structure $\mathcal{A}$ has *local tree-width bounded by g with respect to p* if $\text{ltw}(\mathcal{A}, r) \leq g(r,p)$ for every $r \in \mathbb{N}$. This slightly generalizes the usual notion of local tree-width bounded by a *unary* function [13].

3 Some easy positive instances

In this section, let $k \in \mathbb{N}$ and $G = (V, E)$ be a graph.

Definition 1. *G contains a* k-independent-set-matching structure *on vertices $u_1, \ldots, u_k, v_1, \ldots, v_k$ if $u_1, \ldots, u_k, v_1, \ldots, v_k$ are pairwise distinct; for every $i, j \in [k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$; and $\{u_1, \ldots, u_k\}$ is an independent set in G.*

Lemma 1. *Every graph containing a k-independent-set-matching structure has a k-edge induced subgraph.*

Proof: The case for $k = 0$ is trivially true. So assume $k \geq 1$ and G contains a k-independent-set-matching structure on the vertices $u_1, \ldots, u_k,\ v_1, \ldots, v_k$. We choose the maximum $k' \leq k$ such that $\ell := \Big|E\big(G[\{v_1, \ldots, v_{k'}\}]\big)\Big| \leq k$. If $k' = k$, then $G[V']$ with $V' := \big\{u_1, \ldots, u_{k-\ell}\big\} \cup \big\{v_1, \ldots, v_k\big\}$ is a k-edge induced subgraph of G. Otherwise, $k' < k$. In particular, $\Big|E\big(G[\{v_1, \ldots, v_{k'}, v_{k'+1}\}]\big)\Big| > k$. As $v_{k'+1}$ can contribute at most k' many new edges, we have $\ell + k' > k$, i.e., $k - \ell < k'$. Then $G[V']$ with $V' := \big\{u_1, \ldots, u_{k-\ell}\big\} \cup \big\{v_1, \ldots, v_{k'}\big\}$ is a k-edge induced subgraph of G. □

Definition 2. *G contains a* k-clique-matching structure *on vertices $u_1, \ldots, u_k$, $v_1, \ldots, v_k$ if $u_1, \ldots, u_k,\ v_1, \ldots, v_k$ are pairwise distinct; for every $i, j \in [k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$; and $\{u_1, \ldots, u_k\}$ is a clique in G.*

Lemma 2. *If G contains a k-clique-matching structure, then G has a k-edge induced subgraph.*

Proof: The cases for $k \leq 2$ are trivial. So we consider $k \geq 3$. Let k_0 be maximum with $\binom{k_0}{2} \leq k$ and set $r := k - \binom{k_0}{2}$. It is easy to verify that $k \geq k_0 + r$ by $k \geq 3$ and $k_0 > r$. Now assume G contains a k-clique-matching-structure on the vertices $u_1, \ldots, u_k,\ v_1, \ldots, v_k$. Then, we choose the maximum $r' \leq r$ such that $\ell := \Big|E\big(G[\{v_1, \ldots, v_{r'}\}]\big)\Big| \leq r$. If $r' = r$, then $G[V']$ with $V' := \big\{v_1, \ldots, v_r\big\} \cup \big\{u_1, \ldots, u_{r-\ell}, u_{r+1}, \ldots, u_{k_0+\ell}\big\}$ is a k-edge induced subgraph of G. Otherwise, $r' < r$ and by the maximality of r' we have $\Big|E\big(G[\{v_1, \ldots, v_{r'}, v_{r'+1}\}]\big)\Big| > r$. As $v_{r'+1}$ can add at most r' many new edges, we have $\ell + r' > r$, or equivalently $r - \ell < r'$. It follows that $G[V']$ with $V' := \big\{v_1, \ldots, v_{r'}\big\} \cup \big\{u_1, \ldots, u_{r-\ell}, u_{r'+1}, \ldots, u_{r'+k_0-r+\ell}\big\}$ has exactly k edges. □

Definition 3. *We say that G contains a* k-apex structure *on v_0, A and B if*

(A1) $A, B \subseteq V$ are disjoint with $|A| \geq k$ and $|B| \geq \mathscr{R}_k$, $v_0 \in V$;
(A2) A is a clique in G;
(A3) $\{u, v_0\} \in E$ for every $u \in A$ and $\{v, v_0\} \notin E$ for every $v \in B$;
(A4) $\{u, v\} \in E$ for every $u \in A$ and $v \in B$.

Lemma 3. *If G contains a k-apex structure, then it has a k-edge induced subgraph.*

Proof: The case for $k \leq 1$ is trivially true. So let $k \geq 2$. Moreover, let v_0, A, B be as stated in Definition 3. Since $|B| \geq \mathscr{R}_k$, $G[B]$ contains either a clique of size k or an independent set of size k. If $G[B]$ contains an independent set $B' \subseteq B$ with $|B'| = k$. Then for every $u \in A$ the induced subgraph $G\big[B' \cup \{u\}\big]$ has exactly k edges by (A4). Now assume that there is a clique B' in $G[B]$ of size k. Observe by (A3) and $k \geq 2$, we have $v_0 \notin (A \cup B')$. Furthermore, it is easy to see that we can write $k = \binom{k_0}{2} + r$ for some appropriate $k \geq k_0 \geq r$. We select arbitrary subsets $A' \subseteq A$ and $B'' \subseteq B'$ with $|A'| = r$ and $|B''| = k_0 - r$. Then it is straightforward to check that $G\big[A' \cup B'' \cup \{v_0\}\big]$ has exactly k edges. □

Lemma 4. *Assume there exists three disjoint cliques S_1, S_2, S_3 in G, all of size k; and there are no edges between any distinct S_i and S_j. Then G has a k-edge induced subgraph.*

It is easy to see that Lemma 4 is a direct consequence of Gauss' famous Eureka Theorem [1].

Theorem 1. *For every $k \in \mathbb{N}$ there exist $k_0, k_1, k_2 \in \mathbb{N}$ such that $k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}$.*

Lemma 5. *Let $k \in \mathbb{N}^+$ and $G = (V, E)$ be a graph without isolated vertices. If G contains an independent set of size $(k-1)^2 + 1$, then it has a k-edge induced subgraph.*

To prove the above lemma, we need some further preparation.

Lemma 6. *Let $m, n \in \mathbb{N}^+$ and $A, B \subseteq V$ be disjoint. If for every $u \in A$ we have $\big|N(u) \cap B\big| \geq 1$ and $|A| > (m-1)(n-1)$, then*

(i) either there are m vertices $u_1, \ldots, u_m$ in A, and a vertex v in B with $\{u_i, v\} \in E$ for every $i \in [m]$,

(ii) or there are n vertices $u_1, \ldots, u_n$ in A and n vertices $v_1, \ldots, v_n$ in B such that for all $i, j \in [n]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Proof: [of Lemma 5] Let $S \subseteq V$ be an independent set in G with $|S| > (k-1)^2$. Since G has no isolated vertex, $|N(u) \cap N(S)| \geq 1$ for every $u \in S$. So we can apply Lemma 6 on $A \leftarrow S$, $B \leftarrow N(S)$, $m \leftarrow k$, and $n \leftarrow k$. If (i) holds, then we have an induced k-star of exactly k edges. Otherwise, we have (ii). Hence, there exist vertices $u_1, \ldots, u_k \in S$ and $v_1, \ldots, v_k \in N(S)$ such that G contains a k-independent-set-matching structure on those vertices. The result follows from Lemma 1. □

Definition 4. *Let $d \in \mathbb{N}$. We define*

$$V^G_{[1,d]} := \big\{v \in V \mid 1 \leq \deg(v) \leq d\big\}.$$

It is well-known that if a graph contains many small-degree vertices, then it has a large independent set. As a result, the following is an easy consequence of Lemma 5.

Lemma 7. *Let $d, k \in \mathbb{N}^+$. If $\left|V^G_{[1,d]}\right| > (d+1) \cdot (k-1)^2$, then G contains a k-edge induced subgraph.*

3.1 A further combinatorial lemma

For later purpose, we need a generalization of Lemma 6.

Lemma 8. *Let $m, n, p \in \mathbb{N}^+$ and $A, B \subseteq V$ be disjoint in the graph G. If for every $u \in A$, $\big|N(u) \cap B\big| \geq p$ and $|A| > (m-1)(n-1)^p$, then*

(i) either there are m vertices $u_1, \ldots, u_m$ in A and p vertices $v_1, \ldots, v_p$ in B with $\{u_i, v_j\} \in E$ for every $i \in [m]$ and $j \in [p]$,

(ii) or there are n vertices $u_1, \ldots, u_n$ in A and n vertices $v_1, \ldots, v_n$ in B such that for all $i, j \in [n]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

4 Easy instances by model-checking

In this section we show the fixed-parameter tractability of p-Edge-Induced-Subgraph on some restricted classes of graphs via the model-checking problem for first-order logic. The following is a generalization of a well-known result due to Frick and Grohe [13].

Theorem 2. *For every computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the problem*

> p-Mc-Ltw$_g$-FO
> *Instance:* *A structure $\mathcal{A}$, $p \in \mathbb{N}$ and an FO-sentence φ such that $\mathcal{A}$ has local tree-width bounded by g with respect to p.*
> *Parameter:* $p + |\varphi|$.
> *Problem:* *Decide whether $\mathcal{A} \models \varphi$.*

is fixed-parameter tractable.

Definition 5. *Let $d \in \mathbb{N}$ and $G = (V, E)$ be a graph. If* $\deg(v) \leq d$ *or* $\deg(v) \geq |V| - 1 - d$ *for every $v \in V$, then the graph G is* d-degree-extreme.

Proposition 1. *Let $D : \mathbb{N} \to \mathbb{N}$ be a computable function. Then the problem*

> *Instance:* A graph G and $k \in \mathbb{N}$ such that G is $D(k)$-degree-extreme.
> *Parameter:* k.
> *Problem:* Decide whether G contains a k-edge induced subgraph.

is fixed-parameter tractable.

Definition 6. *Let $d, b \in \mathbb{N}$. Then (G, V_1, V_2, B) is a* (d, b)-bridge *(of the two degree-extreme graphs) if :*

(B1) $V = V_1 \cup V_2$ for some disjoint V_1 and V_2. (B2) $G[V_1]$ and $G[V_2]$ are both d-degree-extreme.
(B3) B is a subset of V with $|B| = b$ such that for every edge $\{u, v\}$ with $u \in V_1$ and $v \in V_2$ we have either $u \in B$ or $v \in B$.

Similar to Proposition 1, we can prove:

Proposition 2. *Let $D : \mathbb{N} \to \mathbb{N}$ be a computable function. Then the problem*

> *Instance:* A graph $G = (V, E)$, $V_1, V_2, B \subseteq V$ and $k \in \mathbb{N}$ such that (G, V_1, V_2, B) is a $(D(k), |B|)$-bridge.
> *Parameter:* $k + |B|$.
> *Problem:* Decide whether G contains a k-edge induced subgraph.

is fixed-parameter tractable.

5 The algorithm

The main component of our fpt-algorithm for p-EDGE-INDUCED-SUBGRAPH is the following procedure that either already solves the problem or decomposes the given graph into potentially a bridge of two large degree-extreme graphs (cf. Definition 6).

For every $k \in \mathbb{N}$ we let $p_k := 2^{2\cdot k}(> \mathscr{R}_k)$.

Lemma 9. *For every computable function $D : \mathbb{N} \to \mathbb{N}$ there is an* fpt-*algorithm $\mathbb{A}_D$ such that for every graph $G = (V, E)$ and every $k \in \mathbb{N}$ exactly one of following conditions is satisfied.*

(S1) G is $D(k)$-degree-extreme and $\mathbb{A}_D$ correctly decides whether G contains a k-edge induced subgraph.
(S2) G is not $D(k)$-degree-extreme and $\mathbb{A}_D$ correctly outputs that G contains a k-edge induced subgraph.
(S3) G is not $D(k)$-degree-extreme and $\mathbb{A}_D$ outputs three subsets $V_1, V_2, B \subseteq V$ such that
(S3.1) $V = V_1 \dot\cup V_2$ with $|V_1| > D(k)$ and $|V_2| > D(k) + 1$;
(S3.2) every edge between V_1 and V_2 in G has one vertex in B and $|B| \leq (p_k - 1)^{p_k+1} + (p_k - 1)^2$.

Proof: Let $G = (V, E)$ be a graph and $k \in \mathbb{N}$. If G is $D(k)$-degree-extreme, then we apply Proposition 1 to achieve (S1). Otherwise let $v_0 \in V$ be a vertex with

$$D(k) < \deg(v_0) < |V| - 1 - D(k). \tag{1}$$

Then we set $V_1 := N(v_0)$ and $V_2 := V \setminus V_1$. By (1) it holds that $|V_1| > D(k)$ and $|V_2| = |V| - |V_1| = |V| - \deg(v_0) > D(k) + 1$, i.e., (S3.1). Let $W_1 := \Big\{u \in V_1 \;\Big|\; \big|N(u) \cap V_2\big| \geq p_k\Big\}$ and $W_2 := V_1 \setminus W_1$.

Claim 1. If $|W_1| > (p_k - 1)^{p_k+1}$, then G contains a k-edge induced subgraph.
Proof of the claim. We apply Lemma 8 on $A \leftarrow W_1$, $B \leftarrow V_2$, $m \leftarrow p_k$, $n \leftarrow p_k$, and $p \leftarrow p_k$. So there are p_k vertices $u_1, \ldots, u_{q_k}$ in W_1 and p_k vertices $v_1, \ldots, v_{p_k}$ in V_2 such that

(i) either $\{u_i, v_j\} \in E$ for every $i, j \in [p_k]$,
(ii) or for all $i, j \in [p_k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Recall $p_k > \mathscr{R}_k$, so there is a subset $S \subseteq \{u_1, \ldots, u_{p_k}\}$ such that S is either an independent set or a clique. If S is an independent set, then $G\big[S \cup \{v_0\}\big]$ has exactly k edges. So suppose S is a clique. Assume that (i) is true, then G contains a k-apex structure on $v_0, S, \{v_1, \ldots, v_{p_k}\}$. Hence, Lemma 3 implies the claim. Otherwise (ii) holds. And say $S = \{u_{i_1}, \ldots, u_{i_k}\}$. Then the graph G contains a k-clique-matching structure on $u_{i_1}, \ldots, u_{i_k}, v_1, \ldots, v_k$. The result follows from Lemma 2. ⊣

Claim 2. If $\big|N(W_2) \cap V_2\big| > (p_k - 1)^2$, then G contains a k-edge induced subgraph.
Proof of the claim. It is easy to verify that we can apply Lemma 6 on $A \leftarrow N(W_2) \cap V_2$, $B \leftarrow W_2$, $m \leftarrow p_k$, and $n \leftarrow p_k$. So,

(i) either there are p_k vertices $u_1, \dots, u_{q_k}$ in $N(W_2) \cap V_2$ and a vertex v in W_2 such that $\{u_i, v\} \in E$ for every $i \in [p_k]$,

(ii) or there are p_k vertices $u_1, \dots, u_{p_k}$ in $N(W_2) \cap V_2$ and p_k vertices $v_1, \dots, v_{p_k}$ in W_2 such that for all $i, j \in [p_k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

But (i) contradicts our definition of W_2, i.e., for every $u \in W_2$ we have $\big|N(u) \cap V_2\big| < p_k$, therefore (ii) must hold. Recall $p_k > \mathscr{R}_k$, hence $G\big[\{v_1, \dots, v_{p_k}\}\big]$ contains either a clique of size of k or an independent set of size k. Without loss of generality, let $\{v_1, \dots, v_k\} \subseteq W_2 \subseteq V_1$ be a clique or an independent set.

For the independent set case, as $v_0 \notin V_1$, then $G\big[\{v_0, v_1, \dots, v_k\}\big]$ is a k-induced subgraph. For the clique case, G contains a k-clique-matching structure on $u_1, \dots, u_k, v_1, \dots, v_k$. We are done by Lemma 2. $\dashv$

Let $B := W_1 \cup \big(N(W_2) \cap V_2\big)$. If $|B| > (p_k-1)^{p_k+1} + (p_k-1)^2$, then, by Claim 1 and Claim 2, the graph G contains a k-edge induced subgraph, and (S2) follows. Otherwise $|B| \leq (p_k-1)^{p_k+1} + (p_k-1)^2$. Observe that every edge between V_1 and V_2 has at least one vertex in B. Thus, we achieve (S3) by outputting (V_1, V_2, B). $\square$

Finally we are ready to present our fpt-algorithm for p-Edge-Induced-Subgraph.

Theorem 3. p-Edge-Induced-Subgraph *is fixed-parameter tractable.*

Proof: We define a computable function $D_0 : \mathbb{N} \to \mathbb{N}$ by

$$D_0(k) := 2 \cdot \big((p_k-1)^{p_k+1} + (p_k-1)^2\big) + 2^{2\cdot((k-1)^2+1)}. \tag{2}$$

Then let $\mathbb{A}_{D_0}$ be the algorithm as stated in Lemma 9 for the function D_0.

Let (G, k) with $G = (V, E)$ be an instance of p-Edge-Induced-Subgraph. First, we remove all the isolated vertices in G. For simplicity, the resulting graph is denoted by G again. Then, we simulate the algorithm $\mathbb{A}_{D_0}$ on (G, k). If the result is either (S1) or (S2) in Lemma 9, we already get the correct answer. Otherwise, $\mathbb{A}_{D_0}$ outputs three subsets $V_1, V_2, B \subseteq V$ satisfying (S3.1) and (S3.2).

If $G[V_1]$ and $G[V_2]$ are both $D_0(k)$-degree-extreme, then (G, V_1, V_2, B) is a $(D_0(k), |B|)$-bridge with $|B|$ bounded by an appropriate computable function of k. The fixed-parameter tractability of whether G contains a k-edge induced subgraph follows from Proposition 2. Otherwise, either $G[V_1]$ or $G[V_2]$ is not $D_0(k)$-degree-extreme.

We assume that $G[V_1]$ is not $D_0(k)$-degree-extreme. (The case for $G[V_2]$ is symmetric.) Then we simulate the algorithm $\mathbb{A}_{D_0}$ on $(G[V_1], k)$. Observe that the result cannot be (S1). If the output is (S2), since $G[V_1]$ is an induced subgraph of G, we conclude that G has an induced subgraph of exactly k edges.

Now we are left with case (S3). In particular, there are subsets $V_{11}, V_{12}, B_1 \subseteq V_1$ such that the corresponding properties of (S3.1) and (S3.2) are satisfied. Let $U_1 := V_{11} \setminus (B \cup B_1)$, $U_2 := V_{12} \setminus (B \cup B_1)$, and $U_3 := V_2 \setminus (B \cup B_1)$. Observe that in G if we remove the vertex set B, then there is no edge left between V_1 and V_2. Similarly, if we remove the vertex set B_1, every edge between V_{11} and V_{12}

is destroyed. Thus, by (S3.2), in the original graph G, there is no edge between each pair of U_1, U_2 and U_3. Moreover by (S3.1) and (S3.2) for every $i \in [3]$

$$|U_i| > D_0(k) - 2 \cdot \left((p_k - 1)^{p_k+1} + (p_k - 1)^2\right) = 2^{2\cdot((k-1)^2+1)} > \mathscr{R}_{(k-1)^2+1},$$

where the equality is by (2).

We use Ramsey's Theorem again. If there is an independent set of size $(k-1)^2+1$ in one of the U_1, U_2 and U_3, as G has no isolated vertex, then G contains a k-edge induced subgraph by Lemma 5. Otherwise every U_i contains a clique of size $(k-1)^2+1 \geq k$. As we have seen that there is no edge between U_1, U_2 and U_3 in G, Lemma 4 implies that G contains an induced subgraph of exactly k edges. □

6 Counting k-edge induced subgraphs

The most natural counting version of p-Edge-Induced-Subgraph is:

> p-#Edge-Induced-Subgraph
> *Instance:* A graph G and $k \in \mathbb{N}$.
> *Parameter:* k.
> *Problem:* Compute the number of k-edge induced subgraphs in G.

In fact, the hardness of p-#Edge-Induced-Subgraph is rather easy to show. We observe that the vertex set of every induced subgraph *without any edge* is an independent set, and vice versa. Hence the *first slice* of p-#Edge-Induced-Subgraph, i.e., counting the number of 0-edge induced subgraphs is exactly the classical problem #Independent-Set of counting the number of independent sets in a given graph. Recall that #Independent-Set is #P-hard [16]. Hence:

Theorem 4. *Assume* #P $\neq$ P*. Then* p-#Edge-Induced-Subgraph *is not fixed-parameter tractable.*

One might attribute the above hardness result to the fact that we allow induced subgraphs to have isolated vertices. Note these isolated vertices play no role in the decision problem p-Edge-Induced-Subgraph. Therefore, it also makes sense to consider:

> p-#Edge-Induced-Subgraph*
> *Instance:* A graph G and $k \in \mathbb{N}$.
> *Parameter:* k.
> *Problem:* Compute the number of k-edge induced subgraphs *without isolated vertices* in G.

Then we can show:

Theorem 5. p-#Edge-Induced-Subgraph* *is hard for* #W[1].

Acknowledgement

We thank Leizhen Cai for bringing the problem p-EDGE-INDUCED-SUBGRAPH to our attention, and Jörg Flum for comments on earlier versions of this paper. This research has been partly supported by the National Nature Science Foundation of China (60970011, 61033002).

References

1. G. Andrews. Eureka! num = $\Delta + \Delta + \Delta$. *Journal of Number Theory*, 23(3):285–293, 1986.
2. H.L. Bodlaender, L. Cai, J. Chen, M.R. Fellows, J.A. Telle, and D. Marx. Open problems in parameterized and exact computation - IWPEC 2006. Technical Report UU-CS-2006-052, Department of Information and Computing Sciences, Utrecht University, 2006.
3. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
4. L. Cai. Private communication, 2008.
5. L. Cai, S.M. Chan, and S.O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proceedings of the second International Workshop on Parameterized and Exact Computation (IWPEC'06)*, Lecture Notes in Computer Science 4169, pages 239–250. Springer, 2006.
6. Y. Chen and J. Flum. On parameterized path and chordless path problems. In *Proceedings of 22nd Annual IEEE Conference on Computational Complexity (CCC 2007)*, pages 250–263. IEEE Computer Society, 2007.
7. Y. Chen, M. Thurley, and M. Weyer. Understanding the complexity of induced subgraph isomorphisms. In *Proceedings of the 35th International Colloquium on Automata Languages and Programming (ICALP'08)*, pages 587–596. Springer, 2008.
8. B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 192–242, Amsterdam, The Netherlands, 1990. Elsevier Science Publishers.
9. R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
10. Z. Dvorak, D. Král, and R. Thomas. Deciding first-order properties for sparse graphs. In *Proceedins of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 133–142. IEEE Computer Society, 2010.
11. J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.
12. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.
13. M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of ACM*, 48(6):1184–1206, 2001.
14. S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.
15. D. Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.
16. L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.