Parameterized Algorithms for Max Colorable Induced Subgraph Problem on Perfect Graphs

Neeldhara Misra[1], Fahad Panolan[2], Ashutosh Rai[2], Venkatesh Raman[2], and Saket Saurabh[2]

[1] Indian Institute of Science, Bangalore, India
neeldhara@csa.iisc.ernet.in
[2] Institute of Mathematical Sciences, Chennai, India
{fahad,ashutosh,vraman,saket}@imsc.res.in

Abstract. We address the parameterized complexity of Max Colorable Induced Subgraph on perfect graphs. The problem asks for a maximum sized q-colorable induced subgraph of an input graph G. Yannakakis and Gavril [*IPL 1987*] showed that this problem is NP-complete even on split graphs if q is part of input, but gave a $n^{O(q)}$ algorithm on chordal graphs. We first observe that the problem is W[2]-hard parameterized by q, even on split graphs. However, when parameterized by ℓ, the number of vertices in the solution, we give two fixed-parameter tractable algorithms.

- The first algorithm runs in time $5.44^{\ell}(n+\#\alpha(G))^{O(1)}$ where $\#\alpha(G)$ is the number of maximal independent sets of the input graph.
- The second algorithm runs in time $q^{\ell+o(\ell)}n^{O(1)}T_{\alpha}$ where T_{α} is the time required to find a maximum independent set in any induced subgraph of G.

The first algorithm is efficient when the input graph contains only polynomially many maximal independent sets; for example split graphs and co-chordal graphs. The running time of the second algorithm is FPT in ℓ alone (whenever T_{α} is a polynomial in n), since $q \leq \ell$ for all non-trivial situations. Finally, we show that (under standard complexity-theoretic assumptions) the problem does not admit a polynomial kernel on split and perfect graphs in the following sense:

(a) On split graphs, we do not expect a polynomial kernel if q is a part of the input.
(b) On perfect graphs, we do not expect a polynomial kernel even for fixed values of $q \geq 2$.

1 Introduction

A fundamental class of graph optimization problems involve finding a maximum induced subgraph satisfying specific properties, such as being edgeless (maximum independent set) [4,5,6,19], acyclic [9], bipartite [4,5], regular [12] or q-colorable [1,20] (equivalent to finding a maximum independent set when $q = 1$, and a maximum induced bipartite subgraph when $q = 2$). Several of these problems are NP-hard on general undirected graphs. Therefore, studies of these problems have involved algorithmic paradigms designed to cope with NP-hardness,

A. Brandstädt, K. Jansen, and R. Reischuk (Eds.): WG 2013, LNCS 8165, pp. 370–381, 2013.

like approximation and parameterization [4,5,9,6,19,20]. The focus of this paper is the Max q-Colorable Induced Subgraph problem, with a special focus on co-chordal graphs and perfect graphs. Our results are of a parameterized flavor, involving both FPT algorithms and lower bounds for polynomial kernels.

Before we can describe our results, we establish some basic notions. A graph $G = (V, E)$ is called *q-colorable* if there is a coloring function $f : V \to [q]$ such that $f(u) \neq f(v)$ for any $(u, v) \in E$. Equivalently, a graph is q-colorable if its vertex set can be partitioned into q independent sets. The Max q-Colorable Induced Subgraph asks for a maximum induced subgraph that is q-colorable, and the decision version, p-mcis, may be stated as follows:

> p-Max Colorable Induced Subgraph (p-mcis) **Parameter:** ℓ
> **Input:** An undirected graph $G = (V, E)$ and positive integers q and ℓ.
> **Question:** Does there exist $Z \subseteq V$, $|Z| \geq \ell$, such that $G[Z]$ is q-colorable?

We will sometimes be concerned with the problem above for *fixed values of* q, and to distinguish this from the case when q is a part of the input, we use p-q-mcis to refer to the version where q is fixed. The problem is clearly NP-complete on general graphs as for $q = 1$ this corresponds to Independent Set problem. Yannakakis and Gavril [20] showed that this problem is NP-complete even on split graphs (which is a proper subset of perfect graphs, chordal graphs and co-chordal graphs, see Section 2 for definitions). However, they showed that p-q-mcis is solvable in time $n^{O(q)}$ on chordal graphs. A natural question, therefore, is whether the problem admits an algorithm with running time $f(q) \cdot n^{O(1)}$ on chordal graphs, or even on split graphs. This question was our main motivation for looking at p-mcis on special graph classes like co-chordal and perfect graphs.

Our study of p-mcis involves determining the parameterized complexity of the problem. The goal of parameterized complexity is to find ways of solving NP-hard problems more efficiently than brute force: here the aim is to restrict the combinatorial explosion to a parameter that is hopefully much smaller than the input size. Formally, a *parametrization* of a problem is assigning an integer k to each input instance and we say that a parameterized problem is *fixed-parameter tractable (*FPT*)* if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function depending on the parameter k only. Just as NP-hardness is used as evidence that a problem probably is not polynomial time solvable, there exists a hierarchy of complexity classes above FPT, and showing that a parameterized problem is hard for one of these classes gives evidence that the problem is unlikely to be fixed-parameter tractable. The principal analogue of the classical intractability class NP is W[1]. A convenient source of W[1]-hardness reductions is provided by the result that Independent Set parameterized by solution size is complete for W[1]. Other highlights of the theory include that Dominating Set, by contrast, is complete for W[2]. For more background, the reader is referred to the monographs [8]. A parameterized problem is said to admit a *polynomial kernel* if every instance (I, k) can be reduced in polynomial time to an equivalent instance with both size and parameter value bounded by a polynomial in k. The study of kernelization is a major research frontier of parameterized complexity and many important recent advances in the area are on kernelization. The recent development of a framework for ruling out polynomial kernels under certain complexity-theoretic assumptions [3,7,10] has added a new dimension to the field and strengthened its connections to classical complexity. For overviews of kernelization we refer to surveys [2,11] and to the corresponding chapters in books on parameterized complexity [8,18].

Our results and related work. Most of the "induced subgraph problems" are known to be W-hard parameterized by the solution size on general graphs by a generic result of Khot and Raman [14]. In particular this also implies that p-MCIS is W[1]-hard parameterized by the solution size on general graphs. Observe that Independent Set is essentially p-MCIS with $q = 1$. There has been also some study of parameterized complexity of Independent Set on special graph classes [6,19]. Yannakakis and Gavril [20] showed that p-MCIS is NP-complete on split graphs and Addario-Berry et al. [1] showed that the problem is NP-complete on perfect graphs for every fixed $q \geq 2$. We observe in passing that the known NP-completeness reduction given in [20] implies that p-MCIS when parameterized by q alone is W[2]-hard even on split graphs. Our main contributions in this paper are two randomized FPT algorithms for p-MCIS and a complementary lower bound, which establishes the non-existence of a polynomial kernel under standard complexity-theoretic assumptions.

Our first algorithm runs in time $(2e)^\ell(n + \#\alpha(G))^{O(1)}$ where $\#\alpha(G)$ is the number of maximal independent sets of the input graph and the second algorithm runs in time $q^\ell \cdot T_\alpha \cdot n^{O(1)}$, where T_α is the time required to compute the largest independent set in any subgraph of the given graph. Observe that since $q \leq \ell$ for all non-trivial situations, we have that the second algorithm is FPT in ℓ alone, provided T_α is a polynomial in n. The first algorithm is efficient when the input graph contains only polynomially many maximal independent sets; for example on split graphs and co-chordal graphs. The second algorithm is efficient for a larger class of graphs, because it only relies on an efficient procedure for finding a maximum independent set (although this comes at the cost of the running time depending on q in the base of the exponent). In particular, the second algorithm runs in time $q^\ell n^{O(1)}$ on the class of perfect graphs. We also describe de-randomization procedures. While the derandomization technique for the first algorithm is standard, to derandomize the second algorithm we need a notion which generalizes the idea of "universal sets", introduced by Naor et al. [16]. We believe that our construction, though simple, could be of independent interest. Further, we show that unless co-NP $\subseteq$ NP/poly, the problem does not admit polynomial kernel even on split graphs. Also, on perfect graphs, we show that the problem does not admit a polynomial kernel even for fixed $q \geq 2$, unless co-NP $\subseteq$ NP/poly.

2 Preliminaries and Definitions

For a finite set V, a pair $G = (V, E)$ such that $E \subseteq V^2$ is a graph on V. The elements of V are called *vertices*, while pairs of vertices (u, v) such that $(u, v) \in E$

are called *edges*. We also use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of G, respectively. In the following, let $G = (V, E)$ and $G' = (V', E')$ be graphs, and let $U \subseteq V$ be some subset of vertices of G. Let G' be a subgraph of G. If E' contains all the edges $\{u, v\} \in E$ with $u, v \in V'$, then G' is an *induced subgraph* of G, *induced by* V', denoted by $G[V']$. For any $U \subseteq V$, we denote $G[V \setminus U]$ by $G \setminus U$. For $v \in V$, $N_G(v) = \{u \mid (u, v) \in E\}$. The complement of a graph $G = (V, E)$, denoted by $\bar{G}$, is the graph with vertex set V and edge set $V \times V \setminus (E \cup \{(v, v) \mid v \in V\})$. A set $X \subseteq V$ is called a clique (resp., independent set) if every pair of vertices in X is adjacent (resp., non-adjacent) in G. X is called a *maximal* clique (resp., independent set), if no proper super set of X is clique (resp., independent set). We denote the size of the maximum clique in graph G by $w(G)$. A graph G is q-colorable if we can partition the vertex set in to q independent sets. The chromatic number $\chi(G)$ of a graph G is the minimum q such that G is q-colorable.

A graph G is called *perfect*, if $\forall\ U \subseteq V(G)$, $w(G[U]) = \chi(G[U])$. A graph $G = (V, E)$ is called *chordal* if every simple cycle of with more than three vertices has an edge connecting two nonconsecutive vertices on the cycle. A graph is *co-chordal* if its complement is a chordal graph. All chordal graphs and co-chordal graphs are perfect graphs. A *split graph* is a graph whose vertex set can be partitioned into two subsets I and Q such that I is an independent set and Q is a clique. Split graphs are closed under complementation. We denote the set $\{1, 2, \ldots, n\}$ by $[n]$ and all possible subsets of size k of $[n]$ by $\binom{[n]}{k}$.

Definition 2.1. *Let $G = (V, E)$ and $H_x = (V_x, E_x)$ for $x \in V$ be graphs. We define the graph $G' = Embed(G; (H_x)_{x\in V})$ as the graph obtained from G by replacing each vertex x with the graph H_x . Formally, $V(G') = \{u_x | x \in V,\ u \in V_x\}$ and $E(G') = \{(u_x, v_x) | (u, v) \in E_x\} \cup \{(u_x, v_y) | (x, y) \in E,\ u \in V_x,\ v \in V_y\}$.*

We say that the graph $Embed(G;\ (H_x)_{x\in V})$ is obtained by embedding $(H_x)_{x\in V}$ into G. We say that a graph class Π is closed under embedding if whenever $G \in \Pi$ and $H_x \in \Pi$, $\forall x \in V(G)$, then the graph $Embed(G;\ (H_x)_{x\in V(G)})$ belongs to Π. It is known that perfect graphs are closed under embedding [15]. Let $G = (V, E)$ be a graph and $E' \subseteq E$. We define the graph $\Delta(G; E')$ as adding vertices x_e and edges (x_e, u), (x_e, v) for all $(u, v) = e \in E'$.

Lemma 2.1 ($\star$). *If $G = (V, E)$ is a perfect graph and $E' \subseteq E$, then $\Delta(G; E')$ is also a perfect graph.*

Due to space constraints, some proofs have been deferred to a full version of the paper. Results whose proofs are omitted are marked with a $\star$.

3 Generalized Universal Sets

In this section we generalize a derandomization tool, *universal sets* given by Naor et al. [16].

Definition 3.1. *An (n, k, q)-universal set is a set of vectors $V \subseteq [q]^n$ such that for any index set $S \in \binom{[n]}{k}$, the projection of V on S contains all possible q^k configurations.*

Theorem 3.1 ($\star$). *An (n, k, q)-universal set of cardinality $q^k k^{O(\log k)} \log^2 n$ can be constructed deterministically in time $O(q^k k^{O(\log k)} n \log^2 n)$.*

Definition 3.2 ([16]). *Let H be a family of functions from $[n]$ to $[l]$. H is an (n, k, l)-family of perfect hash functions if for all $S \in \binom{[n]}{k}$, there is an $h \in H$ which is one-to-one on S.*

Theorem 3.2 ([16]). *There is a deterministic algorithm with running time $O(e^k k^{O(\log k)} n \log n)$ that constructs an (n, k, k)-family of perfect hash functions $\mathcal{F}$ such that $|\mathcal{F}| = e^k k^{O(\log k)} \log n$.*

4 FPT Algorithms

In this section we design two randomized algorithms for p-MCIS. The first algorithm requires a subroutine that enumerates all maximal independent sets in the input graph and this algorithm is useful only when the input graph has polynomially many maximal independent sets. We can derandomize this algorithm using a (n, ℓ, ℓ)-family of perfect hash functions.

The second algorithm requires a subroutine which computes the *maximum* independent set of any induced subgraph of the input graph. Thus, this algorithm is FPT on all graph classes for which Independent Set is either polynomial time solvable or FPT parameterized by the solution size. We derandomize this algorithm using the (n, ℓ, q)-universal sets described in the previous section.

Notice that the second algorithm is less demanding than the first: we only need to find the largest independent set, rather than enumerating all maximal ones. Thus the second algorithm solves the problem for a larger class of graphs than the first, however, as we will see, the running time is compromised in that a dependence on q creeps into the base of the exponent. In particular, this is why the second algorithm does not render the first obsolete. The first can be thought of as a more efficient algorithm when the class of graphs was restricted further.

Algorithm based on enumerating Maximal Indepenent Sets. Let $\#\alpha(G)$ denote the number of maximal independent sets of G, and $T_{\#\alpha}(G)$ denote the time taken to enumerate the maximal independent sets of a graph G. In this section we give a randomized algorithm with one sided error for p-MCIS that uses all the maximal independent sets in the graph, runs in time $T_{\#\alpha}(G) + 2^\ell (n + \#\alpha)^{O(1)}$, and gives the correct answer with probability at least $e^{-\ell}$. The error is one-sided: if the input instance is No instance, then the algorithm will output No always. Thus, in any graph class where the maximal independent sets can be enumerated in polynomial time, we can solve p-MCIS with constant success probability in $O((2e)^\ell n^{O(1)})$ time.

Algorithm 1. An Algorithm for p-MCIS based on enumerating MIS.

Input: A graph $G = (V, E)$ and positive integers ℓ, q
Output: Yes, if there exists $S \subseteq V$, $|S| = \ell$ and $G[S]$ is q-colorable, No otherwise.

1. Enumerate all maximal independent sets in G. Let $M = \{m_1, m_2, \ldots, m_t\}$ be the set of all maximal independent sets.
2. Construct a split graph $G' = (V \uplus M, E' = \{(v, m_i) |\ m_i \in M,\ v \in V \cap m_i\})$, where $G'[M]$ is a clique.
3. Color each vertex in V with a color from an ℓ-sized set of colors uniformly at random.
4. Merge all vertices in each color class into a single vertex. Formally, replace each color class C_i by a single vertex c_i, and let $N(c_i) = \{u \mid \exists v \in C_i, (u, v) \in E'\}$. Let the graph after contraction be $G^* = (C \uplus M, E^*)$.
5. If there exists a partition of C into q sets $C_1, C_2, \ldots, C_q$ such that for all i, C_i has a common neighbor in M, then output Yes, otherwise output No. (This is based on a Steiner Tree computation with C as terminals, see the proof for a description.)

Lemma 4.1. *Algorithm 1 runs in time $O(2^{\ell} n^{O(1)})$ on graphs where the maximal independent sets can be enumerated in polynomial time. Further, if (G, ℓ, q) is a* Yes *instance of p-MCIS, then Algorithm 1 will output* Yes *with probability at least e^{-l}, otherwise Algorithm 1 will output* No *with probability* 1.

Proof. We first argue the running time bound. Since we assume that maximal independent sets are enumerable in polynomial time, Steps 1—4 are clearly polynomial time. To find the partition in Step 5, we run a Steiner Tree algorithm on the instance with C given as the set of terminals. We claim that a partition of the desired kind exists if and only if there exists a Steiner Tree using at most q additional vertices to connect the terminal set C. First, if the set C can be connected with at most q additional vertices $\{s_1, \ldots, s_q\}$ from M, then notice that the non-terminal vertices in the Steiner Tree constitute a dominating set for C (indeed, any non-dominated vertex c_i is necessarily disconnected from $C \setminus \{c_i\}$). Therefore, $\{N(s_i) \setminus \bigcup_{1 \le j < i} N(s_j) \mid 1 \le i \le q\}$ gives the desired partition. On the other hand, suppose we have a partition of C into q sets $C_1, C_2, \ldots, C_q$ such that for all i, C_i has a common neighbor s_i in M. Note that the set $S := \{s_1, \ldots, s_q\}$ is a Steiner Tree for C: given $x \in C_i$ and $y \in C_j$, the path $(x, s_i), (s_i, s_j), (s_j, y)$ (where $s_i = s_j$ if $i = j$) lies in $C \cup S$. Since finding the optimal Steiner Tree on an instance with k terminals can be done in $O(2^k n^{O(1)})$ time [17], we have that the last step of the algorithm runs in time $O(2^{\ell} n^{O(1)})$.

We now show the correctness of the algorithm whenever the output is positive. Suppose Algorithm 1 outputs Yes. Then there exist q vertices in M that dominates all vertices in C which implies at least one vertex in each color class that is dominated by one or more of these q vertices. In particular, there exists a subset $T \subseteq V$ with ℓ vertices and a subset $S \subseteq M$ with q vertices, such that S dominates T. We argue that $G[T]$ is the desired q-colorable subgraph. Let $T := \{v_1, v_2, \ldots, v_\ell\}$. For each v_i, let $c(v_i)$ be the smallest j for which v_i is dominated by m_j. Notice that c defines a partition of T into q sets. For all

Algorithm 2. An Algorithm for p-MCIS based on finding maximum IS.

Input: A graph $G = (V, E)$ and a positive integers ℓ, q
Output: YES, if there exists $S \subseteq V$, $|S| = l$ and $G[S]$ is q-colorable, NO otherwise.

- Color the graph uniformly at random with q colors. Let C_i be the color classes for $1 \leq i \leq q$.
- Find the maximum independent sets H_i for each C_i.
- If $|\bigcup_{1\leq i\leq q} H_i| \geq \ell$, say YES, otherwise say NO.

$1 \leq j \leq q$, it is clear that $c^{-1}(j)$ is a subset of some maximal independent set, and hence the proposed partition is a proper coloring. Therefore, (G, ℓ, q) is a YES instance of p-MCIS.

We now argue the probability that the algorithm finds a solution given that the input is a YES instance. Let (G, ℓ, q) be a YES instance of p-MCIS, and let $T \subseteq V$ with $|T| = \ell$, be a solution. When we randomly color the vertices, each vertex in T will get different colors with probability $\frac{\ell!}{\ell^\ell} \geq e^{-\ell}$. If T gets different colors then there exists q sets in M which dominate C because there exists a maximal independent set that contains each color class in $G[T]$ (since $G[T]$ is q-colorable). Hence Algorithm 1 will output YES with probability at least e^{-l}. □

We can boost the success probability to a constant by executing Algorithm 1 e^ℓ times, in which case the success probability will be at least $(1 - e^{-\ell})^{e^\ell} \geq \frac{1}{e}$. It is easy to see that we can derandomize the algorithm using a (n, ℓ, ℓ)-family of perfect hash functions (see Theorem 3.2) to obtain a deterministic algorithm with running time $(2e)^\ell \ell^{O(\log \ell)} n^{O(1)}$ for p-MCIS on graph classes for which maximal independent sets can be enumerated in polynomial time. Since the number of maximal cliques in chordal graphs with n vertices is bounded by n and all maximal cliques in chordal graphs can be enumerated in polynomial in n time, the number of independent sets in co-chordal graphs are bounded by linear in n and they can be enumerated in polynomial in n time as well. We therefore have the following corollary:

Corollary 4.1. *p-MCIS can be solved in time $(2e)^\ell \cdot \ell^{O(\log \ell)} n^{O(1)}$ on co-chordal graphs and split graphs.*

Algorithm based on finding a Maximum Independent Set. In Algorithm 2, we describe a randomized polynomial time algorithm which succeeds with probability $q^{-\ell}$ on graph classes where MAXIMUM INDEPENDENT SET can be solved in polynomial time.

Lemma 4.2 ($\star$). *If (G, ℓ, q) is a YES instance of p-MCIS, then Algorithm 2 will output YES with probability $q^{-\ell}$, otherwise Algorithm 2 will output NO with probability 1. The algorithm runs in time $T_\alpha \cdot n^{O(1)}$, where T_α is the time required to find a maximum independent set up to size l in any induced subgraph of G.*

Corollary 4.2. *The problem of finding a ℓ-sized q-colorable subgraph on perfect graphs can be solved in time $q^\ell \ell^{O(\log \ell)} n^{O(1)}$.*

5 Kernelization Lower Bounds

In this section we show that MAX INDUCED BIPARTITE SUBGRAPH (i.e, q=2 in p-MCIS) on perfect graphs and p-MCIS on split graphs do not admit polynomial kernels unless CO-NP $\subseteq$ NP/poly.

Lower bound Machinery We begin by stating some of the known techniques developed for showing some problems do not admit polynomial kernels under standard complexity theoretic assumptions.

Definition 5.1 (Composition [3]). *A composition algorithm (also called OR-composition algorithm) for a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, k), ..., (x_t, k))$, with $(x_i, k) \in \Sigma^* \times \mathbb{N}$ for each $1 \leq i \leq t$, uses time polynomial in $\sum_{i=1}^{t} |x_i| + k$, and outputs $(y, k') \in \Sigma^* \times \mathbb{N}$ with (a) $(y, k') \in \Pi \iff (x_i, k) \in \Pi$ for some $1 \leq i \leq t$ and (b) k' is polynomial in k. A parameterized problem is compositional (or OR-compositional) if there is a composition algorithm for it.*

We define the notion of the *unparameterized version* of a parameterized problem Π. The mapping of parameterized problems to unparameterized problems is done by mapping (x, k) to the string $x\#1^k$, where $\# \in \Sigma$ denotes the blank letter and 1 is an arbitrary letter in Σ. In this way, the unparameterized version of a parameterized problem Π is the language $\tilde{\Pi} = \{x\#1^k | (x, k) \in \Pi\}$. The following theorem yields the desired connection between the two notions.

Theorem 5.1 ([3,10]). *Let Π be a compositional parameterized problem whose unparameterized version $\tilde{\Pi}$ is NP-complete. Then, if Π has a polynomial kernel then* CO-NP $\subseteq$ NP/*poly.*

5.1 Max Induced Bipartite Subgraph on Perfect and Split Graphs

The MAX INDUCED BIPARTITE SUBGRAPH problem is formally given as follows:

MAX INDUCED BIPARTITE SUBGRAPH (p-MIBS)	**Parameter:** k
Input: An undirected graph $G = (V, E)$ and a positive integer k.	
Question: Does there exist $S \subseteq V$ such that $\|S\| = k$ and $G[S]$ is bipartite?	

Here, we show that unless CO-NP $\subseteq$ NP/poly, p-MIBS does not have a polynomial kernel when restricted to perfect graphs. We note that we are dealing here with the case of finding a maximum induced bipartite subgraph in the interest of exposition; a more general result that shows the hardness of finding a maximum induced q-colorable subgraph for any fixed $q \geq 2$ on the class of perfect graphs is described in the full version of this work.

Our result here is established by demonstrating an OR-composition. Let $(G_0, k), (G_1, k), \ldots, (G_{t-1}, k)$ be t instances of p-MIBS, where every G_i is a perfect graph. Notice that we may assume that $t \leq 2^{k \log k + k}$. This is because, by Corollary 4.2, we may solve p-MIBS in time $2^{k \log k + k}$ (note that $q = 2$) on perfect graphs. Therefore, if $t > 2^{k \log k + k}$, then we may solve every instance in time

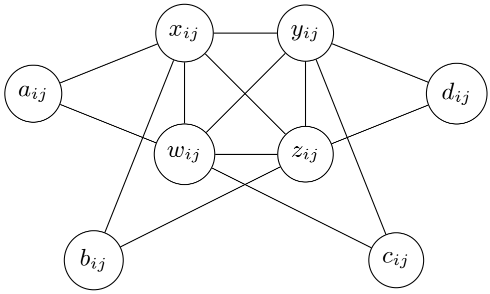

Fig. 1. Identity gadget H_{ij}

$t \cdot 2^{k \log k + k} < t^2$, and return a trivial YES or NO instance as the output of the composition, depending on whether there was at least one YES instance or not, respectively.

Thus, we assume that $t \leq 2^{k \log k + k}$, and therefore, $\log t \leq k^{O(1)}$. For convenience, we assume that t is a power of two (so that $\log t$ is an integral value). This can be done by padding the set of instances with trivial NO instances, and at most doubling the number of instances. We construct a composed instance (G, k^*) as follows. To begin with, let G be the disjoint union of all G_i, $0 \leq i \leq t-1$. For all $i \neq j$ add all possible edges between G_i and G_j.

Now add $2k \log t$ identity gadgets, named H_{ij} for $1 \leq i \leq 2k$, $1 \leq j \leq \log t$. The gadget H_{ij} consists of eight vertices $\{x_{ij}, y_{ij}, w_{ij}, z_{ij}, a_{ij}, b_{ij}, c_{ij}, d_{ij}\}$, where the vertices $\{x_{ij}, y_{ij}, w_{ij}, z_{ij}\}$ form a clique, and the vertex a_{ij} is adjacent to x_{ij} and w_{ij}; b_{ij} is adjacent to x_{ij} and z_{ij}; c_{ij} is adjacent to w_{ij} and y_{ij} and d_{ij} is adjacent to y_{ij} and z_{ij} (see Fig 1). For all $0 \leq l \leq t-1$, if the j^{th} bit of the $\log t$-bit binary representation of l is 0, then add edges from all vertices in G_l to x_{ij} and y_{ij}. Otherwise add edges from all vertices in G_l to w_{ij} and z_{ij}. This completes the description of the composed graph; we let $k^* = k + 12k \log t \leq k + 12k(k + k \log k) = O(k^2 \log k)$. Having shown that k^* is polynomially dependent on k, for simplicity, in the remaining discussion we continue refer to k^* in terms of t. We first show that this is indeed a valid OR-composition, and then demonstrate that G, as described, is a perfect graph.

Lemma 5.1. *The instance $(G, k + 12k \log t)$ is a* YES *instance of p-MIBS if, and only if, (G_l, k) is a* YES *instance of p-MIBS for some $0 \leq l \leq (t-1)$.*

Proof. ($\Rightarrow$) Assume $(G, k + 12k \log t)$ is a YES instance of p-MIBS and let $S \subseteq V(G)$ be a solution. We first claim that S will not contain vertices from more than two input instances. Indeed, suppose not. Then for $i_1 \neq i_2 \neq i_3$, let $v_{i_1} \in S \cap V(G_{i_1})$, $v_{i_2} \in S \cap V(G_{i_3})$ and $v_{i_3} \in S \cap V(G_{i_3})$. Note that $v_{i_1}, v_{i_2}, v_{i_3}$ will induce a triangle and contradict the fact that $G[S]$ is bipartite. We now assume that S contains vertices from two input graphs G_p and G_q. If one of them has at least k vertices in S, then we are done. Otherwise, $|S \cap V(G_p)| + |S \cap V(G_q)| < 2k$. Hence,

$$\sum_{i=1}^{2k}\sum_{j=1}^{\log t} |S \cap V(H_{ij})| > k + 12k\log t - 2k \geq 12k\log t - k$$

Therefore, by an averaging argument, there exists an i' such that $\sum_{j=1}^{\log t} |S \cap V(H_{i'j})| \geq 6\log t$. Since vertices $x_{ij}, y_{ij}, w_{ij}, z_{ij}$ from H_{ij} form a complete graph, S can contain at most 2 vertices from $\{x_{ij}, y_{ij}, w_{ij}, z_{ij}\}$. So $|S \cap V(H_{ij})| \leq 6$ and if $|S \cap V(H_{ij})| = 6$ then either $S \cap V(H_{ij}) = \{a_{ij}, b_{ij}, c_{ij}, d_{ij}, x_{ij}, y_{ij}\}$ or $S \cap V(H_{ij}) = \{a_{ij}, b_{ij}, c_{ij}, d_{ij}, w_{ij}, z_{ij}\}$. We know that to meet the budget, it must be the case that $\forall j,\ |S \cap V(H_{i'j})| = 6$.

Since $p \neq q$ there exists a j' such that j'^{th} bit of binary representation of p and q are different (say 0 and 1, respectively). Hence, all the vertices from G_p are connected to $x_{i'j'}, y_{i'j'}$ and all the vertices from G_q are connected to $w_{i'j'}, z_{i'j'}$. Hence there exists a triangle in $G[S \cap (V(G_p) \cup V(G_q) \cup V(H_{i'j'}))]$. This contradicts the fact that $G[S]$ is bipartite, showing that the case $|S \cap V(G_p)| + |S \cap V(G_q)| < 2k$ is infeasible. The remaining case is when S contains vertices from at most one input graph (say G_p). Since $|S \cap V(H_{ij})| \leq 6$, S will contain at least k vertices from $V(G_p)$. Hence $S \cap V(G_p)$ is a solution of (G_p, k).
($\Leftarrow$) Let (G_p, k) be a YES instance of p-MIBS, and let $S \subseteq V(G_p)$ be the solution. Let $b_1b_2\ldots b_{\log t}$ be the binary representation of p. Now consider the vertex set

$$\begin{aligned} T := \{x_{ij}, y_{i,j} \mid 1 \leq i \leq 2k \ \wedge\ b_j = 1\} \cup \{w_{ij}, z_{i,j} \mid 1 \leq i \leq 2k \ \wedge\ b_j = 0\} \\ \cup \{a_{ij}, b_{ij}, c_{ij}, d_{ij} | 1 \leq i \leq 2k \ \wedge\ 1 \leq j \leq \log t\}. \end{aligned} \tag{1}$$

It is easy to see that T involves exactly six vertices from each of the $2k\log t$ gadgets, and the vertices are chosen such that $G[T]$ induces a bipartite graph. Further, the vertices are chosen to ensure that there are no edges between vertices in S and vertices in T, and therefore, it is clear that $G[S \cup T]$ induces a bipartite subgraph of G of the desired size. Hence $(G, k + 12k\log t)$ is a YES instance of p-MIBS. □

Lemma 5.2. *The graph G constructed as the output of the OR-composition is a perfect graph.*

Proof. We begin by describing an auxiliary graph G', and show that G' is perfect. This graph is designed to be a graph from which G can be obtained by a series of operations that preserve perfectness, and this will lead us to establishing that G is perfect. The graph G' contains a clique on t vertices, K_t. We let $V(K_t) := \{v_0, v_1, \ldots v_{t-1}\}$. G' also contains $2k\log t$ small graphs, each of which consist of two vertices with an edge between them (i.e, each small graph is an edge). Let $\{n_{ij}, p_{ij}\}$ for all $1 \leq i \leq 2k$, $1 \leq j \leq \log t$ be the vertices of small graphs. For all $0 \leq l \leq t-1$, if the j^{th} bit of the $\log t$-bit binary representation of l is 0, then add edges from v_l to n_{ij} for all i. Otherwise add edges from v_l to p_{ij} for all i.

We claim that G' is perfect. Let H be an induced subgraph of G'. If $|V(H) \cap V(K_t)| \leq 1$, then H is a forest and so in this case $\omega(H) = \chi(H)$. Else, $r =$

$|V(H) \cap V(K_t)| \geq 2$. Since the neighborhoods of n_{ij} and p_{ij} do not intersect, and there are no edges between small graphs in G', at most one vertex from the entire set of small graphs can be part of the largest clique in H containing $V(H) \cap V(K_t)$ (note that there exists a largest clique that contains all the vertices in $V(H) \cap V(K_t)$). So $\omega(H) \leq r+1$. Let us denote by H^* the subgraph $H[V(H) \cap \{n_{ij}, p_{ij} \mid 1 \leq i \leq 2k, 1 \leq j \leq \log t\}]$.

If $\omega(H) = r+1$, then we define the following coloring. Color all r vertices in $V(H) \cap V(K_t)$ with colors $1, 2, \ldots, r$. For all $x \in V(H^*)$ such that x is adjacent to all vertices in $V(H) \cap V(K_t)$, we give a color $r+1$ (note that these vertices are independent by construction). If an $x \in V(H^*)$ is not adjacent to all vertices in $V(H) \cap V(K_t)$, then we can color it with a color that is already used on one of its non adjacent vertices in $V(H) \cap V(K_t)$. If $\omega(H) = r$, then there is no vertex in $V(H^*)$ which is adjacent to $V(H) \cap V(K_t)$. So we can color vertices in $V(H) \cap V(K_t)$ with r colors and for a vertex $x \in V(H^*)$ we can color x with a color same as (one of) its non adjacent vertex in $V(H) \cap V(K_t)$.Hence $\omega(H) = \chi(H)$.

Let be G^* be a graph obtained by embedding G_i on $v_i \in V(G')$ for all $0 \leq i \leq t-1$ and embedding an edge on each vertex in $\{n_{ij}, p_{ij} \mid 1 \leq i \leq 2k, 1 \leq j \leq \log t\}$. It can be observed that G^* is isomorphic to

$$G \setminus \bigcup_{1 \leq i \leq 2k, 1 \leq j \leq \log t} \{a_{ij}, b_{ij}, c_{ij}, d_{ij}\}.$$

It follows that G^* is perfect. Finally, observe that the graph G is $\Delta(G^*; E')$ for a suitable choice of $E' \subseteq E(G^*)$, and it follows that G is perfect. □

Lemmas 5.1,5.2 and Theorem 5.1, give us the following result.

Theorem 5.2. p-MAX INDUCED BIPARTITE SUBGRAPH *on perfect graphs does not admit a polynomial kernel unless* co-NP $\subseteq$ NP/*poly.*

We finally show that p-MCIS does not admit a polynomial kernel on split graphs unless co-NP $\subseteq$ NP/poly by showing a "parameter-preserving reduction" from SMALL UNIVERSE SET COVER.

Theorem 5.3 ($\star$). p-MCIS *on split graphs does not admit a polynomial kernel unless* co-NP $\subseteq$ NP/*poly.*

6 Conclusion

In this paper we studied the parameterized complexity of p-MCIS on perfect graphs and showed that the problem is FPT when parameterized by the solution size. We also studied its kernelization complexity and showed that the problem does not admit polynomial kernel under certain complexity theory assumptions. An interesting direction of research that this paper opens up is the study of parameterized complexity of INDUCED SUBGRAPH ISOMORPHISM on special graph classes. As a first step it would be interesting to study the parameterized complexity of INDUCED TREE ISOMORPHISM parameterized by the size of the tree on perfect graphs.

References

1. Addario-Berry, L., Kennedy, W.S., King, A.D., Li, Z., Reed, B.A.: Finding a maximum-weight induced k-partite subgraph of an i-triangulated graph. Discrete Applied Mathematics 158(7), 765–770 (2010)
2. Bodlaender, H.L.: Kernelization: New upper and lower bound techniques. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 17–37. Springer, Heidelberg (2009)
3. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. 75(8), 423–434 (2009)
4. Byskov, J.M.: Algorithms for k-colouring and finding maximal independent sets. In: SODA, pp. 456–457 (2003)
5. Byskov, J.M.: Enumerating maximal independent sets with applications to graph colouring. Oper. Res. Lett. 32(6), 547–556 (2004)
6. Dabrowski, K., Lozin, V.V., Müller, H., Rautenbach, D.: Parameterized algorithms for the independent set problem in some hereditary graph classes. In: Iliopoulos, C.S., Smyth, W.F. (eds.) IWOCA 2010. LNCS, vol. 6460, pp. 1–9. Springer, Heidelberg (2011)
7. Dell, H., van Melkebeek, D.: Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In: STOC, pp. 251–260 (2010)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag New York, Inc., Secaucus (2006)
9. Fomin, F.V., Gaspers, S., Pyatkin, A.V., Razgon, I.: On the minimum feedback vertex set problem: Exact and enumeration algorithms. Algorithmica 52(2), 293–307 (2008)
10. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. In: STOC, pp. 133–142 (2008)
11. Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. SIGACT News 38(1), 31–45 (2007)
12. Gupta, S., Raman, V., Saurabh, S.: Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. SIAM J. Discrete Math. 26(4), 1758–1780 (2012)
13. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: On generating all maximal independent sets. Inf. Process. Lett. 27(3), 119–123 (1988)
14. Khot, S., Raman, V.: Parameterized complexity of finding subgraphs with hereditary properties. Theor. Comput. Sci. 289(2), 997–1008 (2002)
15. Lovasz, L.: Perfect graphs. In: Beineke, L.W., Wilson, R.J. (eds.) Selected Topics in Graph Theory, vol. 2, pp. 55–67. Academic Press, London (1983)
16. Naor, M., Schulman, L.J., Srinivasan, A.: Splitters and near-optimal derandomization. In: FOCS, pp. 182–191 (1995)
17. Nederlof, J.: Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 713–725. Springer, Heidelberg (2009)
18. Niedermeier, R.: Invitation to Fixed Parameter Algorithms. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, USA (2006)
19. Raman, V., Saurabh, S.: Short cycles make w-hard problems hard: Fpt algorithms for w-hard problems in graphs with no short cycles. Algorithmica 52(2), 203–225 (2008)
20. Yannakakis, M., Gavril, F.: The maximum k-colorable subgraph problem for chordal graphs. Inf. Process. Lett. 24(2), 133–137 (1987)