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] The 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 second algorithm is FPT in ℓ alone, 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 empty (maximum independent set) [7,8,9,28], acyclic [14], bipartite [7,8,29], regular [20] or q-colorable [1,30] (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, like approximation and parameterization [7,8,14,9,28,30]. 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 $Z \subseteq V$, $\lvert Z\rvert \geq \ell$, 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 [30] 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). 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 $f(k) \cdot \lvert I\rvert^{O(1)}$, where $\lvert I\rvert$ 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 [12,13,27]. 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. These include general results showing that certain classes of parameterized problems have polynomial kernels [2,5,15] or randomized kernels [23]. The recent development of a framework for ruling out polynomial kernels under certain complexity-theoretic assumptions [4,10,16] has added a new dimension to the field and strengthened its connections to classical complexity. For overviews of kernelization we refer to surveys [3,19] and to the corresponding chapters in books on parameterized complexity [13,27].

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 [22]. 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 [9,28]. Yannakakis and Gavril [30] 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 [30] 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. 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. [25]. 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

Graphs. 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 $U \subseteq V$ 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 $U \subseteq V$, $G \setminus U = G[V \setminus U]$. For $v \in V$, $N_G(v) = \{u \mid (u, v) \in E\}$. Complement of a simple undirected 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/independent set if every pair of vertices in X is adjacent/non-adjacent in G. X is called a *maximal* clique/independent set, if no proper super set of X is clique/independent set. We denote size of maximum clique in graph G by $w(G)$. 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. Complement of chordal graphs are called *co-chordal* graphs. 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 [24].

Definition 2.2. *Let $G = (V, E)$ be a graph and $E' \subseteq E$. We define the graph $Triangular(G;\ E')$ as adding vertices x_e and edges $(x_e, u),\ (x_e, v)$ for all $(u, v) = e \in E'$.*

If $G = (V, E)$ is a perfect graph and $E' \subseteq E$, then $Triangular(G;\ E')$ is also a perfect graph (the proof of this statement is implicit in the proof of Theorem 5.3).

3 Generalized universal sets

In this section we generalize a derandomization tool, *universal sets* given by Naor et al. [25].

3.1 Definitions

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.*

We can also look at a vector $v \in [q]^n$ as a function from $[n]$ to $[q]$. We will be using the two notions interchangeably. An (n, k, q)-universal set is a special case of *k-restriction problem* introduced by Naor et al. [25].

k-RESTRICTION PROBLEM	
Input:	Positive integers b, k, n and a list $\mathcal{C} = C_1, C_2, ..., C_m$ where $C_i \subseteq [b]^k$ and with the collection $\mathcal{C}$ being invariant under permutation of $[k]$
Output:	Collection of vectors $\mathcal{V} \subseteq [b]^n$ such that $\forall\ S \in \binom{[n]}{k}$ and $\forall j \in [m]$, $\exists v \in \mathcal{V}$ such that projection of v on S, $v(S) \in C_j$.

To specify (n, k, q)-universal set as *k-restriction problem*, let $b = q$ and $\mathcal{C}$ consist of $C_x = \{x\}$ for all $x \in [q]^k$.

Theorem 3.1 **([25]).** *For any k-restriction problem with* $b \le n$*, there is a deterministic algorithm that outputs a collection obeying the k-restrictions, with the size*

$$t = \lceil \frac{k \ln n + \ln m}{\ln(b^k/(b^k - c))} \rceil, \text{ where } c = \min_{1 \le j \le m} |C_j| \tag{1}$$

The time taken to output the collection is

$$O\left(\frac{b^k}{c}\binom{n}{k} mTn^k\right) \tag{2}$$

where T *is the time complexity to check whether or not* $v(S) \in C_j$*, given* $v \in [b]^n$*,* $S \in \binom{[n]}{k}$ *and* $j \in [m]$*,*

By substituting values for (n, k, q) universal sets we get the following corollary.

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

Definition 3.2 **([25]).** *An* (n, k, l)-splitter H *is a family of functions from* $[n]$ *to* $[l]$ *such that for all* $S \in \binom{[n]}{k}$*, there is an* $h \in H$ *that splits* S *perfectly, i.e., into equal sized parts* $h^{-1}(j) \cap S$*,* $j = 1, 2, ..., l$ *(or as equal as possible, if* l *does not divide* k*).*

Definition 3.3 **([25]).** *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*.*

We need the following result regarding (n, k, k)-family of perfect hash functions.

Theorem 3.2 **([25]).** *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* $|\mathcal{F}| = e^k k^{O(\log k)} \log n$.

Theorem 3.3 **([17]).** *An* (n, k, k^2)*-family of perfect hash functions of size* $O(\frac{k^4 \log^2 n}{\log(k \log n)})$ *can be constructed in time* $O(n \cdot poly(k, \log n))$.

Lemma 3.1 **([25]).** *For any* $k \leq n$ *and for all* $l \leq n$*, there is an explicit family* $B(n, k, l)$ *of* (n, k, l)*-splitters of size* $\binom{n}{l-1}$

3.2 Efficient Construction of (n, k, q)-universal sets

Naor et al. [25] gave a construction of (n, k, q) universal sets for $q = 2$, which has size $O(2^k k^{O(\log k)} \log n)$ and can be listed in time linear in the size of universal sets. In this subsection, we generalize the result for $q \geq 2$.
Construction. Let $l = c \log k$ for some constant c (we will fix c later). Let $A = A(n, k, k^2)$, $B = B(k^2, k, l)$ and $C = C(k^2, k/l, q)$ be respective function families presented by Theorem 3.3, Lemma 3.1 and Corollary 3.1. Then our required (n, k, q)-universal sets is a family of functions H,

$$H = \{(a, b, c_1, c_2, ...c_l) | a \in A, b \in B, \forall i \in [l] : c_i \in C\}$$

where each $(a, b, c_1, c_2, ..., c_l) \in H$ is defined by

$$(a, b, c_1, c_2, ..., c_l)(x) = c_{b(a(x))}\,(a(x))$$

Correctness. It can be easily verified that each $h \in H$ maps $[n]$ to $[q]$. Let $S \in \binom{[n]}{k}$. We need to show that the restriction of functions in H on S gives all possible functions from $S \to [q]$. By the property of A there exist a function $a \in A$ which is *one-to-one* on S. Let $S' = \{i \mid \exists j \in S : a(j) = i\}$. Since a is *one-to-one* on S, $|S'| = k$. By the property of B, there exist $b \in B$ such that b splits S' equally into l blocks. Let $S'_i = \{j \mid j \in S' \text{ and } b(j) = i\}$ for all $i \in [l]$. Now by the property of C, we have restriction of functions from C on S'_i is all possible functions from $S'_i \to [q]$, Combining these functions from C with a and b, we get all possible functions from $S \to [q]$.
Size and Time. From Theorem 3.3, we know that $|A| = O\left(\frac{k^4 \log^2 n}{\log(k \log n)}\right)$ and A can be constructed in time $O(n \cdot poly(k, \log n))$. By Lemma 3.1, $|B| = \binom{k^2}{l-1} = k^{O(\log k)}$ and B can be constructed in time $k^{O(\log k)}$. By Corollary 3.1, $|C| = O(q^{k/l} k \log k)$ and can be constructed in time $q^{O(k/l)} 2^{(4k \log k)/l}$ which is at most $q^{c_1 k/(c \log k)} 2^{4k/c}$ for some fixed constant c_1. Since we know that $q \geq 2$, we can choose a large enough constant c such that the running time becomes $O(q^k)$. Now, size of the family H is,

$$\begin{aligned} |H| &= |A| \times |B| \times |C|^l \\ &= O\left(\frac{k^4 \log^2 n}{\log(k \log n)}\right) \cdot k^{O(\log k)} \cdot (q^k k^{O(\log k)}) \\ &= q^k k^{O(\log k)} \log^2 n \end{aligned}$$

Also, we see that the time taken for constructing the sets A, B and C is at most the size of H (or within a constant factor). Hence, H can be constructed in time linear in its size. Thus we get the following theorem.

Theorem 3.4. *An (n, k, q)-universal set of cardinality $q^k k^{O(\log k)} \log^2 n$ can be constructed deterministically in time $O(q^k k^{O(\log k)} n \log^2 n)$.*

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 doesn't render the first obsolete. The first can be thought of as a more efficient algorithm when the class of graphs was restricted further.

4.1 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.

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. For instance, on the class of split graphs or co-chordal graphs. It is

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.

worth to mention that maximal independent sets can be enumerated in polynomial delay on general graphs [21]. 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 [26], 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 $1 \le j \le q$, 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

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.

$T \subseteq V$ 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}. $\square$

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.*

4.2 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. *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.*

Proof. It suffices to show if (G, ℓ, q) is a YES instance, then the algorithm outputs YES with probability at least q^{-l}, since it is clear that when Algorithm 2 outputs YES, then (G, ℓ, q) is indeed a YES instance. Let (G, ℓ, q) be a YES instance of p-MCIS, $S \subseteq V, |S| = \ell$ be the solution set and $f : S \to [q]$ be a fixed proper coloring of $G[S]$. We call a coloring $g : V \to [q]$ of G *good* if for all $v \in S$,

$g(v) = f(v)$. Now, since we are choosing the colors for each vertex uniformly at random, the probability that the vertex set S gets exactly the same color given by the function f is $1/q$. Hence, the probability of a random coloring being a *good* coloring is $(1/q)^\ell$. Let $S_i = \{v : v \in S, f(v) = i\}$ and $C_i = \{v : v \in V, g(v) = i\}$, where g is the random coloring in the algorithm. Clearly, if g is a good coloring, then $S_i \subseteq C_i$ for all $1 \leq i \leq q$, and the maximum independent set, H_i in C_i is at least as large as S_i. The correctness follows. The running time is apparent, since Steps 1 and 3 are clearly polynomial time, and Step 2 invokes an algorithm for finding a Maximum Independent Set (up to size ℓ) q times, which takes time $T_\alpha \cdot q$. □

Notice that in particular, if G belongs to a hereditary graph class where computing the maximum independent set can be done in polynomial time, then Algorithm 2 runs in polynomial time. On the other hand, suppose we have an algorithm that finds an independent set of size at most k in time $f(k)n^{O(1)}$. Then, for $1 \leq i \leq q$, we run this algorithm in $G[C_i]$, for values of k in the range $1, 2, \ldots, \ell - (\sum_{j=0}^{i-1} X_j)$, where X_j is the largest independent set in H_i and $X_0 := 0$. We move from $G[C_i]$ to $G[C_{i+1}]$ when the algorithm returns No for the first time. We stop when we exhaust the range, as that is precisely when we have identified a subgraph on ℓ vertices with the desired property. Notice that the FPT algorithm is called at most ℓ times, and the maximum value of the parameter used is ℓ, therefore the overall running time is bounded by $f(\ell)n^{O(1)}$.

As before, if we repeat the above algorithm q^ℓ times, we can solve p-MCIS on perfect graphs with constant success probability in $O(q^\ell)n^{O(1)}$ time. We can derandomize this algorithm using (n, ℓ, q)-universal sets. We know that there exist (n, ℓ, q) universal sets of size $q^\ell \ell^{O(\log \ell)} \log^2 n$ (see section 3). Combined with the fact that maximum independent sets can be found in polynomial time on perfect graphs [18], this brings us to the final consequence in this section:

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. In Subsection 5.1, we state the known lower bound machinery to rule out polynomial kernels and in the subsequent subsections we apply the lower bound machinery to the above problems.

5.1 Lower bound Machinery

In this subsection, we state some of the known techniques developed for showing some problems do not admit polynomial kernels under some complexity theoretic assumptions.

Definition 5.1 (Composition [4]). *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 ([4,16]). *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.*

For some problems, obtaining a composition algorithm directly is a difficult task. Instead, we can give a reduction from a problem that provably has no polynomial kernel unless co-NP $\subseteq$ NP/poly to the problem in question such that a polynomial kernel for the problem considered would give a kernel for the problem we reduced from. We now define the notion of polynomial parameter transformations.

Definition 5.2 ([6]). *Let P and Q be parameterized problems. We say that P is polynomial parameter reducible to Q, written $P \leq_{ppt} Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ and a polynomial p, such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ (a) $(x, k) \in P \iff (x', k') = f(x, k) \in Q$ and (b) $k' \leq p(k)$. The function f is called polynomial parameter transformation.*

Proposition 5.1 ([6]). *Let P and Q be the parameterized problems and $\tilde{P}$ and $\tilde{Q}$ be the unparameterized versions of P and Q respectively. Suppose that $\tilde{P}$ is NP-complete and $\tilde{Q}$ is in NP. Furthermore if $P \leq_{ppt} Q$, then if Q has a polynomial kernel then P also has a polynomial kernel.*

5.2 Max Induced Bipartite Subgraph on Perfect 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

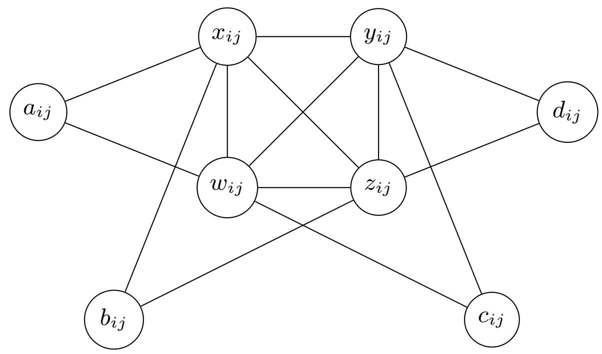

Fig. 1. Identity gadget H_{ij}

a maximum induced q-colorable subgraph for any fixed $q \geq 2$ on the class of perfect graphs is described in the next section.

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 $t \cdot 2^{k \log k + k} < t^2$, 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 $t = 2^{k \log k + k}$ (note that we assume equality without loss of generality, since whenever t falls short, the set of instances can be padded with trivial NO 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 $k^* = k + 12k \log t = k + 12k(k + k \log k) = O(k^2 \log k)$. 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. (⇒) 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,

$$\sum_{i=1}^{2k} \sum_{j=1}^{\log t} |S \cap V(H_{ij})| > k + 12k \log t - 2k \geq 12k \log t - k \tag{3}$$

Therefore 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).
(⇐) Let (G_p, k) be a YES instance of p-MIBS, and let $S \subseteq V(G_p)$ be the solution. Let $b_1 b_2 \ldots b_{\log t}$ be the binary representation of p. Now consider the vertex set

$$\begin{aligned} T := \{x_{ij}, y_{i,j} \mid 1 \leq i \leq 2k \ \wedge\ b_j = 1\} \cup \{w_{ij}, z_{i,j} \mid 1 \leq i \leq 2k \ \wedge\ b_j = 0\} \\ \cup \{a_{ij}, b_{ij}, c_{ij}, d_{ij} | 1 \leq i \leq 2k \ \wedge\ 1 \leq j \leq \log t\}. \end{aligned} \tag{4}$$

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 $1 \le i \le 2k$, $1 \le j \le \log t$ be the vertices of small graphs. For all $0 \le l \le t-1$, 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 the G' is perfect. Let H be an induced subgraph of G'. If $|V(H) \cap V(K_t)| \le 1$, then H is a forest and so in this case $\omega(H) = \chi(H)$. Otherwise $r = |V(H) \cap V(K_t)| \ge 2$. 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 $\omega(H) \le r+1$. Let us denote by H^* the subgraph $H[V(H) \cap \{n_{ij}, p_{ij} \mid 1 \le i \le 2k, 1 \le j \le \log t\}]$.

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 $0 \le i \le t-1$ and embedding an edge on each vertex in $\{n_{ij}, p_{ij} \mid 1 \le i \le 2k, 1 \le j \le \log t\}$. It can be observed that G^* is isomorphic to

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

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

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.*

5.3 *p*-*q*-mcis on Perfect Graphs

We now show the hardness of finding a maximum induced q-colorable subgraph for any fixed $q \ge 2$ on the class of perfect graphs.

Theorem 5.3. p-mcis *for a fixed q on perfect graphs does not admit polynomial kernel unless* co-NP $\subseteq$ NP/*poly*

Proof. We prove the theorem using OR-composition. Let the input instances of OR-composition algorithm be $(G_0, k), (G_1, k), \ldots, (G_{t-1}, k)$. Now we construct an instance (G, k) as follows. For all $i \neq j$ add all possible edges between G_i and G_j. Now add $2qk \log t$ identity gadgets, named H_{ij} for $1 \leq i \leq 2qk,\ 1 \leq j \leq \log t$, as follows. Each H_{ij} contain two cliques K^0_{ij} and K^1_{ij} of size q each. We add all possible edges between K^0_{ij} and K^1_{ij}. Let

$$I_{ij} = \{(C_0, C_1) \mid C_0 \subseteq V(K^0_{ij}), C_1 \subseteq V(K^1_{ij}), |C_0| + |C_1| = q, 0 < |C_0|, |C_1| < q\}$$

Now for each $(X, Y) \in I_{ij}$ we add a vertex $v_{X,Y}$ to H_{ij} and add edges $\{(v_{X,Y}, u) \mid (X, Y) \in I_{ij}, u \in X \vee u \in Y\}$. Let $V_{ij} = \{v_{X,Y} \mid (X, Y) \in I_{ij}\}$ and $p = |I_{ij}| = |V_{ij}|$. Note that p is a function of q only. 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 all vertices in K^0_{ij} for all i. Otherwise add edges from all vertices in G_l to all vertices in K^1_{ij} for all i. The graph G, so far constructed along with parameter $k + 2qk(p + q) \log t$ is the output of the OR-composition algorithm.

Now we show that G is a perfect graph. Consider the following graph G'. G' contains a clique on t vertices, K_t. Let the vertices of K_t are named $v_0, v_1, \ldots v_{t-1}$. G' also contains $2qk \log t$ small graphs, on two vertices and one edge each (i.e each small graph is an edge). Let n_{ij}, p_{ij} for all $1 \leq i \leq 2qk,\ 1 \leq j \leq \log t$ 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. Using similar arguments in the proof of lemma 5.2 we can show that G' is a perfect graph. 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 a clique of size q on each vertex in $\{n_{ij}, p_{ij} \ :\ \text{for all } i, j\}$. So G'' is a perfect graph. It can be observed that G'' is isomorphic to $G \setminus \bigcup_{ij} V_{ij}$. We claim that if $G'' = (V, E)$ is perfect graph and $X \subseteq V$ such that $G[X]$ is a clique, then the graph $G^* = (V \cup \{u\}, E \cup \{(u, x) \mid x \in X\})$ is a perfect graph. Let H be an induced subgraph of G^*. If $u \notin V(H)$, then $w(H) = \chi(H)$ because H is an induced subgraph of a perfect graph G''. Now consider the case $u \in V(H)$. We know that $w(H \setminus \{u\}) = \chi(H \setminus \{u\})$. Let $d = w(H \setminus \{u\}) = \chi(H \setminus \{u\})$. Since $G[N_H(u)]$ is a clique, $d \geq N_H(u)$. If $d > N_H(u)$, the largest clique size in H will be d and we can color H with d colors by giving color to u which is not the color of any of its neighbors. So $w(H) = \chi(H)$. If $d = N_H(u)$, the largest clique size in H is $d+1$ ($G[u \cup N_H(u)]$) and we can color H using $d+1$ colors by giving a new color to u. Hence G^* is a perfect graph. Note that we can get graph G (we constructed for OR-composition) by repeatedly applying the above operation on G'' using vertices from $\bigcup_{ij} V_{ij}$. Therefore G is a perfect graph.

Now we show that $(G, k + 2qk(p+q) \log t)$ is a Yes instance of p-MCIS if and only if $\exists l$ such that (G_l, k) is a Yes instance of p-MCIS.
($\Leftarrow$) Let (G_l, k) be a Yes instance of p-MCIS. Let $S \subseteq V(G_l)$ be the solution. Let $b_1 b_2 \ldots b_{\log t}$ be the binary representation of l. Now consider the vertex set

$$T = \bigcup_{ij} \left(V \left(K^{1-b_j}_{ij} \right) \cup V_{ij} \right).$$

It is easy to see that $|T| = 2qk(p+q)\log t$. We claim that $G[T]$ is q-colorable. For that it is enough to show that $G[T \cap V(H_{ij})]$ is q-colorable because there are no edges between identity gadgets. Consider $G[T \cap V(H_{ij})]$ for any fixed i, j. Let $\{k_1, k_2, \ldots, k_q\} = V(K_{ij}^{1-b_j})$. We keep each k_r in color class r. Since for each $v_{X,Y} \in V_{ij}$, $\exists\, k_s \in V(K_{ij}^{1-b_j})$ such that $(k_s, v_{X,Y}) \notin E(G)$, we can keep $v_{X,Y}$ in color class s. Also note that V_{ij} form an independent set. Hence $G[T \cap V(H_{ij})]$ is q-colorable. Since there is no edges between S and T, $G[S \cup T]$ is a q-colorable induced subgraph of G, of size $k + 2qk(p+q)\log t$.
($\Rightarrow$) Assume $(G, k + 2qk(p+q)\log t)$ is a YES instance of q-MCIS. Let $S \subseteq V(G)$ be the solution set. We claim S will not contain vertices from more than q input instances. Suppose not, then S will contain a $q+1$ clique, which contradict the fact that $G[S]$ is q-colorable. So assume that S contain vertices from at most q input graphs $G_{i_1}, \ldots, G_{i_h}$ where $h \le q$. If one of them has at least k vertices in S, then we are done. Otherwise $\sum_{j=1}^{h} |S \cap V(G_{i_j})| < qk$. Hence

$$\sum_{i,j} |S \cap V(H_{ij})| \ge k + 2qk(p+q)\log t - qk \tag{5}$$

$$\ge 2qk(p+q)\log t - (q-1)k \tag{6}$$

Therefore $\exists\, i'$ such that $\sum_{j=1}^{\log t} |S \cap V(H_{i'j})| \ge (p+q)\log t$. Since $G[V(K_{ij}^0) \cup V(K_{ij}^1)]$ is a complete graph, S can contain at most q vertices from $V(K_{ij}^0) \cup V(K_{ij}^1)$. So $|S \cap V(H_{ij})| \le (p+q)$. If $|S \cap V(H_{ij})| = (p+q)$ then either $S \cap V(H_{ij}) = V(K_{ij}^0) \cup V_{ij}$ or $S \cap V(H_{ij}) = V(K_{ij}^1) \cup V_{ij}$ because if S contain $q_1 (> 0)$ vertices from $V(K_{ij}^0)$ and $q_2 (> 0)$ vertices from $V(K_{ij}^1)$, then there exist a vertex in V_{ij} that can not be part of S. Hence $\forall j, \quad |S \cap V(H_{i'j})| = (p+q)$ and $\forall j$, $V(K_{ij}^0) \subseteq S \cap V(H_{i'j})$ or $V(K_{ij}^1) \subseteq S \cap V(H_{i'j})$, but not both. Since $i_1 \neq i_2$ there exists a j' such that j'^{th} bit of binary representation of i_1 and i_2 are different (say it is 0 and 1 resp.). Hence all the vertices from G_{i_1} is connected to $V(K_{ij}^0)$ and all the vertices from G_{i_2} is connected to $V(K_{ij}^0)$. Hence there exist a $q+1$ sized clique in $G[S \cap (V(G_{i_1}) \cup V(G_{i_2}) \cup V(H_{i'j'}))]$. It contradict the fact that $G[S]$ is q-colorable. Therefore S contain vertices from one input graph (say G_l) only and since $|S \cap V(H_{ij})| \le (p+q)$, S will contain at least k vertices from $V(G_l)$. Hence $S \cap V(G_l)$ is a solution of (G_l, k). □

5.4 *p*-MCIS on Split Graphs

We now show that p-MCIS does not admit polynomial kernel unless CO-NP $\subseteq$ NP/poly by showing a PPT reduction from SMALL UNIVERSE SET COVER.

SMALL UNIVERSE SET COVER	**Parameter:** n
Input: A set $U = \{u_1, \ldots, u_n\}$, a family $\mathcal{F}$ of subsets of X and an integer k	
Question: Does there exist a subfamily $\mathcal{F}' \in \binom{\mathcal{F}}{k}$ such that $\bigcup_{S \in \mathcal{F}'} S = U$	

We have the following theorem due to Dom et al [11].

Theorem 5.4 ([11]). SMALL UNIVERSE SET COVER *does not admit polynomial kernel unless* CO-NP $\subseteq$ NP/*poly*

In fact Dom et al [11] showed that SMALL UNIVERSE SET COVER parameterized by n and k does not admit polynomial kernel unless co-NP $\subseteq$ NP/poly. Since $k \leq n$ for all non-trivial cases, we have Theorem 5.4.

Lemma 5.3. *There is a polynomial parameter transformation from* SMALL UNIVERSE SET COVER *to* p-MCIS.

Proof. The reduction we give here is along the lines of the NP-Complete reduction for p-MCIS by Yannakakis and Gavril [30]. Given an instance $(U, \mathcal{F}, k)$ of SMALL UNIVERSE SET COVER, we construct an instance (G, l, q) of p-MCIS as follows. The split graph G has vertex set $(X \cup \mathcal{F})$ with X being the independent set, and $\mathcal{F}$ inducing the clique. For any $u \in X$ and $S \in \mathcal{F}$ we add an edge (u, S) if and only if $u \notin S$. We set $l = n + k$ and $q = k$. Since $k \leq n$, $l \leq 2n$.

We claim that $(U, \mathcal{F}, k)$ is a YES instance of SMALL UNIVERSE SET COVER if and only if (G, l, q) is a YES instance of p-MCIS.

Suppose $(U, \mathcal{F}, k)$ is a YES instance of SMALL UNIVERSE SET COVER and let $S_1, S_2, \ldots, S_k$ be a solution. The graph induced on $X \cup \{S_1, S_2, \ldots, S_k\}$ is k colorable because the vertex S_i with its non-neighbors in X (they are exactly the elements in the set S_i) form an independent set. Suppose, on the other hand, that (G, l, q) is a YES instance of p-MCIS. Let H be a q-colorable subgraph of G. Since vertices in $\mathcal{F}$ form a clique, $|V(H) \cap \mathcal{F}| = k$ and so let $\{S_1, \ldots, S_k\} = V(H) \cap \mathcal{F}$. Hence $X \subseteq V(H)$. Let $V_1, \ldots, V_k$ be the q color classes in H. Since $S_1, \ldots, S_k$ form a clique, for all $i \neq j$, S_i and S_j will be in two different color classes. Now it is easy to see that corresponding sets $S_1, \ldots, S_k$ cover U, because for each $u \in U$, u is covered by S_i where $u, S_i \in V_j$ for some j. □

Using Proposition 5.1, Theorem 5.4 and Lemma 5.3, we get the following

Theorem 5.5. 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. L. Addario-Berry, W. S. Kennedy, A. D. King, Z. Li, and B. A. Reed. Finding a maximum-weight induced k-partite subgraph of an i-triangulated graph. *Discrete Applied Mathematics*, 158(7):765–770, 2010.

2. N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo. Solving MAX-r-SAT above a tight lower bound. In *SODA*, pages 511–517, 2010.
3. H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, pages 17–37, 2009.
4. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
5. H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (meta) kernelization. In *FOCS*, pages 629–638, 2009.
6. H. L. Bodlaender, S. Thomassé, and A. Yeo. Analysis of data reduction: Transformations give evidence for non-existence of polynomial kernels. Technical report, 2008.
7. J. M. Byskov. Algorithms for k-colouring and finding maximal independent sets. In *SODA*, pages 456–457, 2003.
8. J. M. Byskov. Enumerating maximal independent sets with applications to graph colouring. *Oper. Res. Lett.*, 32(6):547–556, 2004.
9. K. Dabrowski, V. V. Lozin, H. Müller, and D. Rautenbach. Parameterized algorithms for the independent set problem in some hereditary graph classes. In *IWOCA*, volume 6460 of *Lecture Notes in Computer Science*, pages 1–9. Springer-Verlag, 2010.
10. H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260, 2010.
11. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and ids. In *ICALP (1)*, pages 378–389, 2009.
12. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999. 530 pp.
13. J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
14. F. V. Fomin, S. Gaspers, A. V. Pyatkin, and I. Razgon. On the minimum feedback vertex set problem: Exact and enumeration algorithms. *Algorithmica*, 52(2):293–307, 2008.
15. F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *SODA*, pages 503–510, 2010.
16. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142, 2008.
17. M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.
18. M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
19. J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
20. S. Gupta, V. Raman, and S. Saurabh. Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. *SIAM J. Discrete Math.*, 26(4):1758–1780, 2012.
21. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.
22. S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theor. Comput. Sci.*, 289(2):997–1008, 2002.
23. S. Kratsch and M. Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In *SODA*, pages 94–103, 2012.

24. L. Lovasz. Perfect graphs. In L. W. Beineke and R. J. Wilson, editors, *Selected Topics in Graph Theory, Volume 2*, pages 55–67. Academic Press, London-New York, 1983.
25. M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *FOCS*, pages 182–191, 1995.
26. J. Nederlof. Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In *ICALP (1)*, pages 713–725, 2009.
27. R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006.
28. V. Raman and S. Saurabh. 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.
29. V. Raman, S. Saurabh, and S. Sikdar. Efficient exact algorithms through enumerating maximal independent sets and other techniques. *Theory Comput. Syst.*, 41(3):563–587, 2007.
30. M. Yannakakis and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. *Inf. Process. Lett.*, 24(2):133–137, Jan. 1987.