Parameterized Complexity of Independent Set in H-Free Graphs

Édouard Bonnet
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Nicolas Bousquet
CNRS, G-SCOP laboratory, Grenoble-INP, France

Pierre Charbit
Université Paris Diderot, IRIF, France

Stéphan Thomassé
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France
Institut Universitaire de France

Rémi Watrigant
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Abstract

In this paper, we investigate the complexity of Maximum Independent Set (MIS) in the class of H-free graphs, that is, graphs excluding a fixed graph as an induced subgraph. Given that the problem remains NP-hard for most graphs H, we study its fixed-parameter tractability and make progress towards a dichotomy between FPT and $W[1]$-hard cases. We first show that MIS remains $W[1]$-hard in graphs forbidding simultaneously $K_{1,4}$, any finite set of cycles of length at least 4, and any finite set of trees with at least two branching vertices. In particular, this answers an open question of Dabrowski *et al.* concerning C_4-free graphs. Then we extend the polynomial algorithm of Alekseev when H is a disjoint union of edges to an FPT algorithm when H is a disjoint union of cliques. We also provide a framework for solving several other cases, which is a generalization of the concept of *iterative expansion* accompanied by the extraction of a particular structure using Ramsey's theorem. Iterative expansion is a maximization version of the so-called *iterative compression*. We believe that our framework can be of independent interest for solving other similar graph problems. Finally, we present positive and negative results on the existence of polynomial (Turing) kernels for several graphs H.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Parameterized Algorithms, Independent Set, H-Free Graphs

Digital Object Identifier 10.4230/LIPIcs.IPEC.2018.17

Related Version A full version of the paper is available at [4], `https://arxiv.org/abs/1810.04620`.

Funding É. B. is supported by the LABEX MILYON (ANR-10- LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). N. B. and P. C. are supported by the ANR Project DISTANCIA (ANR-17-CE40-0015) operated by the French National Research Agency (ANR).

13th International Symposium on Parameterized and Exact Computation (IPEC 2018).
Editors: Christophe Paul and Michał Pilipczuk; Article No. 17; pp. 17:1–17:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Given a simple graph G, a set of vertices $S \subseteq V(G)$ is an *independent set* if the vertices of this set are all pairwise non-adjacent. Finding an independent set with maximum cardinality is a fundamental problem in algorithmic graph theory, and is known as the MIS problem (MIS, for short) [12]. In general graphs, it is not only NP-hard, but also not approximable within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless $P = NP$ [20], and $W[1]$-hard [10] (unless otherwise stated, n always denotes the number of vertices of the input graph). Thus, it seems natural to study the complexity of MIS in restricted graph classes. One natural way to obtain such a restricted graph class is to forbid some given pattern to appear in the input. For a fixed graph H, we say that a graph is H*-free* if it does not contain H as an induced subgraph. Unfortunately, it turns out that for most graphs H, MIS in H-free graphs remains NP-hard, as shown by a very simple reduction first observed by Alekseev:

▶ **Theorem 1** ([1]). *Let H be a connected graph which is neither a path nor a subdivision of the claw. Then* MIS *is NP-hard in H-free graphs.*

On the positive side, the case of P_t-free graphs has attracted a lot of attention during the last decade. While it is still open whether there exists $t \in \mathbb{N}$ for which MIS is NP-hard in P_t-free graphs, quite involved polynomial-time algorithms were discovered for P_5-free graphs [17], and very recently for P_6-free graphs [13]. In addition, we can also mention the recent following result: MIS admits a subexponential algorithm running in time $2^{O(\sqrt{tn \log n})}$ in P_t-free graphs for every $t \in \mathbb{N}$ [3]. The second open question concerns the subdivision of the claw. Let $S_{i,j,j}$ be a tree with exactly three vertices of degree one, being at distance i, j and k from the unique vertex of degree three. The complexity of MIS is still open in $S_{1,2,2}$-free graphs and $S_{1,1,3}$-free graphs. In this direction, the only positive results concern some subcases: it is polynomial-time solvable in $(S_{1,2,2}, S_{1,1,3}, dart)$-free graphs [15], $(S_{1,1,3}, banner)$-free graphs and $(S_{1,1,3}, bull)$-free graphs [16], where *dart*, *banner* and *bull* are particular graphs on five vertices. Given the large number of graphs H for which the problem remains NP-hard, it seems natural to investigate the existence of parameterized algorithms[1], that is, determining the existence of an independent set of size k in a graph with n vertices in time $O(f(k)n^c)$ for some computable function f and constant c. A very simple case concerns K_r-free graphs, that is, graphs excluding a clique of size r. In that case, Ramsey's theorem implies that every such graph G admits an independent set of size $\Omega(n^{\frac{1}{r-1}})$, where $n = |V(G)|$. In the FPT vocabulary, it implies that MIS in K_r-free graphs has a kernel with $O(k^{r-1})$ vertices.

To the best of our knowledge, the first step towards an extension of this observation within the FPT framework is the work of Dabrowski *et al.* [8] (see also Dabrowski's PhD manuscript [7]) who showed, among others, that for any positive integer r, Max Weighted Independent Set is FPT in H-free graphs when H is a clique of size r minus an edge. In the same paper, they settle the parameterized complexity of MIS on almost all the remaining cases of H-free graphs when H has at most four vertices. The conclusion is that the problem is FPT on those classes, except for $H = C_4$ which is left open. We answer this question by showing that MIS remains $W[1]$-hard in a subclass of C_4-free graphs. On the negative side, it was proved that MIS remains $W[1]$-hard in $K_{1,4}$-free graphs [14].

Finally, we can also mention the case where H is the *bull* graph, which is a triangle with a pending vertex attached to two different vertices. For that case, a polynomial Turing kernel was obtained [19] then improved [11].

[1] For the sake of simplicity, "MIS" will denote the optimisation, decision and parameterized version of the problem (in the latter case, the parameter is the size of the solution), the correct use being clear from the context.

1.1 Our results

In Section 2, we present three reductions proving $W[1]$-hardness of MIS in graph excluding several graphs as induced subgraphs, such as $K_{1,4}$, any fixed cycle of length at least four, and any fixed tree with two branching vertices. In Section 3, we extend the polynomial algorithm of Alekseev when H is a disjoint union of edges to an FPT algorithm when H is a disjoint union of cliques. In Section 4, we present a general framework extending the technique of *iterative expansion*, which itself is the maximization version of the well-known iterative compression technique. We apply this framework to provide FPT algorithms when H is a clique minus a complete bipartite graph, or when H is a clique minus a triangle. Finally, in Section 5, we focus on the existence of polynomial (Turing) kernels. We first strenghten some results of the previous section by providing polynomial (Turing) kernels in the case where H is a clique minus a claw. Then, we prove that for many H, MIS on H-free graphs does not admit a polynomial kernel, unless $NP \subseteq coNP/poly$. Our results allows to obtain the complete dichotomy polynomial/polynomial kernel (PK)/no PK but polynomial Turing kernel/$W[1]$-hard for all possible graphs on four vertices, while only five graphs on five vertices remain open for the $FPT/W[1]$-hard dichotomy.

Due to space restrictions, proofs marked with a ($\star$) were omitted, and can be found in the long version of the paper [4]. This long version also contains additional figures, and two variants of the reduction presented in the next section, together with a discussion.

1.2 Notation

For classical notation related to graph theory or fixed-parameter tractable algorithms, we refer the reader to the monographs [9] and [10], respectively. For an integer $r \geq 2$ and a graph H with vertex set $V(H) = \{v_1, \dots, v_{n_H}\}$ with $n_H \leq r$, we denote by $K_r \setminus H$ the graph with vertex set $\{1, \dots, r\}$ and edge set $\{ab : 1 \leq a, b \leq r \text{ such that } v_a v_b \notin E(H)\}$. For $X \subseteq V(G)$, we write $G \setminus X$ to denote $G[V(G) \setminus X]$. For two graphs G and H, we denote by $G \uplus H$ the *disjoint union* operation, that is, the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. We denote by $G + H$ the *join* operation of G and H, that is, the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H) \cup \{uv : u \in V(G), v \in V(H)\}$. For two integers r, k, we denote by $Ram(r, k)$ the Ramsey number of r and k, *i.e.* the minimum order of a graph to contain either a clique of size r or an independent set of size k. We write for short $Ram(k) = Ram(k, k)$. Finally, for $\ell, k > 0$, we denote by $Ram_\ell(k)$ the minimum order of a complete graph whose edges are colored with ℓ colors to contain a monochromatic clique of size k.

2 $W[1]$-hardness

▶ **Theorem 2.** *For any $p_1 \geq 4$ and $p_2 \geq 1$,* MIS *remains $W[1]$-hard in graphs excluding simultaneously the following graphs as induced subgraphs: $K_{1,4}$, C_4, …, C_{p_1} and any tree T with two branching vertices*[2] *at distance at most p_2.*

Proof. Let $p = \max\{p_1, p_2\}$. We reduce from Grid Tiling, where the input is composed of k^2 sets $S_{i,j} \subseteq [m] \times [m]$ $(0 \leq i, j \leq k-1)$, called *tiles*, each composed of n elements. The objective of Grid Tiling is to find an element $s^*_{i,j} \in S_{i,j}$ for each $0 \leq i, j \leq k-1$, such that $s^*_{i,j}$ agrees in the first coordinate with $s^*_{i,j+1}$, and agrees in the second coordinate with $s^*_{i+1,j}$,

[2] A branching vertex in a tree is a vertex of degree at least 3.

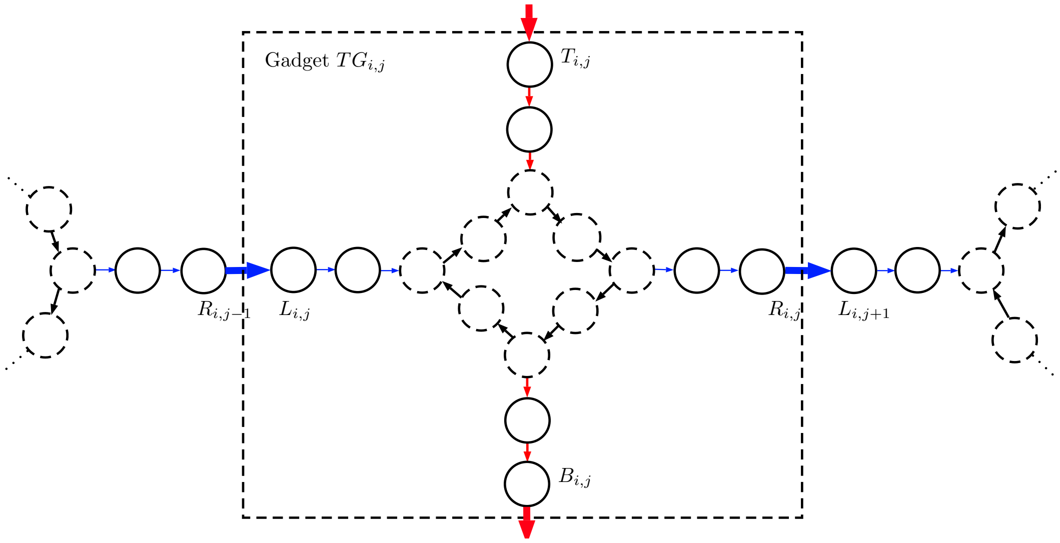

Figure 1 Gadget $TG^{i,j}$ representing a tile and its adjacencies with $TG_{i,j-1}$ and $TG_{i,j+1}$, for $p = 1$. Each circle is a clique on n vertices (dashed cliques are the cycle cliques). Black, blue and red arrows represent respectively type T_h, T_r and T_c edges (bold arrows are between two gadgets).

for every $0 \le i, j \le k-1$ (incrementations of i and j are done modulo k). In such case, we say that $\{s^*_{i,j},\ 0 \le i, j \le k-1\}$ is a *feasible solution* of the instance. It is known that Grid Tiling is $W[1]$-hard parameterized by k [6].

Before describing formally the reduction, let us give some definitions and ideas. Given $s = (a, b)$ and $s' = (a', b')$, we say that s is *row-compatible* (resp. *column-compatible*) with s' if $a \ge a'$ (resp. $b \ge b'$)[3]. Observe that a solution $\{s^*_{i,j},\ 0 \le i, j \le k-1\}$ is feasible if and only if $s^*_{i,j}$ is row-compatible with $s^*_{i,j+1}$ and column-compatible with $s^*_{i+1,j}$ for every $0 \le i, j \le k-1$ (incrementations of i and j are done modulo k). Informally, the main idea of the reduction is that, when representing a tile by a clique, the row-compatibility (resp. column-compatibility) relation (as well at its complement) forms a C_4-free graph when considering two consecutive tiles, and a claw-free graph when considering three consecutive tiles. The main difficulty is to forbid the desired graphs to appear in the "branchings" of tiles. We now describe the reduction.

For every tile $S_{i,j} = \{s^{i,j}_1, \dots, s^{i,j}_n\}$, we construct a *tile gadget* $TG_{i,j}$, depicted in Figure 1. Notice that this gadget shares some ideas with the $W[1]$-hardness of the problem in $K_{1,4}$-free graphs by Hermelin *et al.* [14]. To define this gadget, we first describe an oriented graph with three types of arcs (type T_h, T_r and T_c, which respectively stands for *half graph*, *row* and *column*, this meaning will become clearer later), and then explain how to represent the vertices and arcs of this graph to get the concrete gadget. Consider first a directed cycle on $4p+4$ vertices $c_1, \dots, c_{4p+4}$ with arcs of type T_h. Then consider four oriented paths on $p+1$ vertices: P_1, P_2, P_3 and P_4. P_1 and P_3 are composed of arcs of type T_c, while P_2 and P_4 are composed of arcs of type T_r. Put an arc of type T_c between the last vertex of P_1 and c_1, an arc of type T_c between c_{2p+3} and the first vertex of P_3, an arc of type T_r between c_{p+2} and the first vertex of P_2, and an arc of type T_r between the last vertex of P_4 and c_{3p+4}.

[3] Notice that the row-compatibility (resp. column-compatibility) relation is not symmetrical.

Now, replace every vertex of this oriented graph by a clique on n vertices, and fix an arbitrary ordering on the vertices of each clique. For each arc of type T_h between c and c', add a half graph[4] between the corresponding cliques: connect the a^{th} vertex of the clique representing c with the b^{th} vertex of the clique representing c' iff $a > b$. For every arc of type T_r from a vertex c to a vertex c', connect the a^{th} vertex of the clique representing c with the b^{th} vertex of the clique representing c' iff $s_a^{i,j}$ is *not* row-compatible with $s_b^{i,j}$. Similarly, for every arc of type T_c from a vertex c to a vertex c', connect the a^{th} vertex of the clique representing C with the b^{th} vertex of the clique representing c' iff $s_a^{i,j}$ is *not* column-compatible with $s_b^{i,j}$. The cliques corresponding to vertices of this gadget are called the *main cliques* of $TG_{i,j}$, and the cliques corresponding to the central cycle on $4p+4$ vertices are called the *cycle cliques*. The main cliques which are not cycle cliques are called *path cliques*. The cycle cliques adjacent to one path clique are called *branching cliques*. Finally, the clique corresponding to the vertex of degree one in the path attached to c_1 (resp. c_{p+2}, c_{2p+3}, c_{3p+4}) is called the *top* (resp. *right*, *bottom*, *left*) clique of $TG_{i,j}$, denoted by $T_{i,j}$ (resp. $R_{i,j}$, $B_{i,j}$, $L_{i,j}$). Let $T_{i,j} = \{t_1^{i,j}, \ldots, t_n^{i,j}\}$, $R_{i,j} = \{r_1^{i,j}, \ldots, r_n^{i,j}\}$, $B_{i,j} = \{b_1^{i,j}, \ldots, b_n^{i,j}\}$, and $L_{i,j} = \{\ell_1^{i,j}, \ldots, \ell_n^{i,j}\}$. For the sake of readability, we might omit the superscripts i, j when it is clear from the context.

▶ **Lemma 3.** *($\star$) Let K be an independent set of size $8(p+1)$ in $TG_{i,j}$. Then:*
(a) *K intersects all the cycle cliques on the same index x;*
(b) *if $K \cap T_{i,j} = \{t_{x_t}\}$, $K \cap R_{i,j} = \{r_{x_r}\}$, $K \cap B_{i,j} = \{b_{x_b}\}$, and $K \cap L_{i,j} = \{\ell_{x_\ell}\}$. Then:*
- *$s_{x_\ell}^{i,j}$ is row-compatible with $s_x^{i,j}$ which is row-compatible with $s_{x_r}^{i,j}$, and*
- *$s_{x_t}^{i,j}$ is column-compatible with $s_x^{i,j}$ which is column-compatible with $s_{x_b}^{i,j}$.*

For $i, j \in \{0, \ldots, k-1\}$, we connect the right clique of $TG_{i,j}$ with the left clique of $TG_{i,j+1}$ in a "type T_r spirit": for every $x, y \in [n]$, connect $r_x^{i,j} \in R_{i,j}$ with $\ell_y^{i,j+1} \in L_{i,j+1}$ iff $s_x^{i,j}$ is *not* row-compatible with $s_y^{i,j+1}$. Similarly, we connect the bottom clique of $TG_{i,j}$ with the top clique of $TG_{i+1,j}$ in a "type T_c spirit": for every $x, y \in [n]$, connect $b_x^{i,j} \in B_{i,j}$ with $t_y^{i+1,j} \in T_{i+1,j}$ iff $s_x^{i,j}$ is *not* column-compatible with $s_y^{i+1,j}$ (all incrementations of i and j are done modulo k). This terminates the construction of the graph G.

▶ **Lemma 4.** *($\star$) The input instance of* Grid Tiling *is positive if and only if G has an independent set of size $k' = 8(p+1)k^2$.*

Let us now prove that G does not contain the graphs mentionned in the statement as an induced subgraph:

(i) $K_{1,4}$: we first prove that for every $0 \leq i, j \leq k-1$, the graph induced by the cycle cliques of $TG_{i,j}$ is claw-free. For the sake of contradiction, suppose that there exist three consecutive cycle cliques A, B and C containing a claw. W.l.o.g. we may assume that $b_x \in B$ is the center of the claw, and $a_\alpha \in A$, $b_\beta \in B$ and $c_\gamma \in C$ are the three endpoints. By construction of the gadgets (there is a half graph between A and B and between B and C), we must have $\alpha < x < \gamma$. Now, observe that if $x < \beta$ then a_α must be adjacent to b_β, and if $\beta < x$, then b_β must be adjacent to c_γ, but both case are impossible since $\{a_\alpha, b_\beta, c_\gamma\}$ is supposed to be an independent set. Similarly, we can prove that the graph induced by each path of size $2(p+1)$ linking two consecutive gadgets is claw-free. Hence, the only way for $K_{1,4}$ to appear in G would be that

[4] Notice that our definition of half graph slighly differs from the usual one, in the sense that we do not put edges relying two vertices of the same index. Hence, our construction can actually be seen as the complement of a half graph (which is consistent with the fact that usually, both parts of a half graph are independent sets, while they are cliques in our gadgets).

the center appears in the cycle clique attached to a path, for instance in the clique represented by the vertex c_1 in the cycle. However, it can easily be seen that in this case, a claw must lie either in the graph induced by the cycle cliques of the gadget, or in the path linking $TG_{i,j}$ with $TG_{i-1,j}$, which is impossible.

(ii) $C_4, \ldots, C_{p_1}$. The main argument is that the graph induced by any two main cliques does not contain any of these cycles. Then, we show that such a cycle cannot lie entirely in the cycle cliques of a single gadget $TG_{i,j}$. Indeed, if this cycle uses at most one vertex per main clique, then it must be of length at least $4p+4$. If it intersects a clique C on two vertices, then either it also intersect all the cycle cliques of the gadget, in which case it is of length $4p+5$, or it intersects an adjacent clique of C on two vertices, in which case these two cliques induce a C_4, which is impossible. Similarly, such a cycle cannot lie entirely in a path between the main cliques of two gadgets. Finally, the main cliques of two gadgets are at distance $2(p+1)$, hence such a cycle cannot intersect the main cliques of two gadgets.

(iii) any tree T with two branching vertices at distance at most p_2. Using the same argument as for the $K_{1,4}$ case, observe that the claws contained in G can only appear in the cycle cliques where the paths are attached. However, observe that these cliques are at distance $2(p+1) > p_2$, thus, such a tree T cannot appear in G. ◀

3 Positive results I: disjoint union of cliques

For $r, q \geq 1$, let K_r^q be the disjoint union of q copies of K_r. The following proof is inspired by the case $r = 2$ by Alekseev [2].

▶ **Theorem 5.** Maximum Independent Set *is FPT in K_r^q-free graphs.*

Proof. We will prove by induction on q that a K_r^q-free graph has an independent set of size k or has at most $(Ram(r,k)+1)^{qk}n^{qr}$ independent sets. This will give the desired FPT-algorithm, as the proof shows how to construct this collection of independent sets. Note that the case $q = 1$ is trivial by Ramsey's theorem.

Let G be a K_r^q-free graph and let $<$ be any fixed total ordering of $V(G)$ such that the largest vertex in this ordering belongs to a clique of size r (the case where G is K_r-free is trivial by Ramsey's theorem). For any vertex x, define $x^+ = \{y,\ x < y\}$ and $x^- = V(G) \setminus x^+$ (hence, $x \in x^-$).

Let C be a fixed clique of size r in G and let c be the largest vertex of C with respect to $<$. Let V_1 be the set of vertices of c^+ which have no neighbor in C. Note that V_1 induces a K_r^{q-1}-free graph, so by induction either it contains an independent set of size k, and so does G, or it has at most $(Ram(r,k)+1)^{(q-1)k}n^{(q-1)r}$ independent sets. In the latter case, let $\mathcal{S}_1$ be the set of all independent sets of $G[V_1]$.

Now in a second phase we define an initially empty set $\mathcal{S}_C$ and do the following. For each independent set S_1 in $\mathcal{S}_1$ (including the empty set), we denote by V_2 the set of vertices in c^- that have no neighbor in S_1 (notice that $c \in V_2$). For every choice of a vertex x amongst the largest $(Ram(r,k)+1)$ vertices of V_2 in the order, we add x to S_1 and modify V_2 in order to keep only vertices that are smaller than x (with respect to $<$) and not adjacent to x. We repeat this operation k times (or less if V_2 becomes empty) and, at the end, we either find an independent set of size k or add S_1 to $\mathcal{S}_C$. By doing so we construct a family of at most $(Ram(r,k)+1)^k$ independent sets for each S_1, so in total we get indeed at most $(Ram(r,k)+1)^{kq}n^{(q-1)r}$ independent sets for each clique C. Finally we define $\mathcal{S}$ as the union over all r-cliques C of the sets $\mathcal{S}_C$, so that $\mathcal{S}$ has size at most the desired number.

We claim that if G does not contain an independent set of size k, then $\mathcal{S}$ contains all independent sets of G. It suffices to prove that for every independent set S, there exists a clique C for which $S \in \mathcal{S}_C$. Let S be an independent set, and define C to be a clique of size r such that its largest vertex c (with respect to $<$) satisfies the conditions:

- no vertex of C is adjacent to a vertex of $S \cap c^+$, and
- c is the smallest vertex such that a clique C satisfying the first item exists.

First remark that such a clique always exist, since we assumed that the largest vertex c_{last} of $<$ is contained in a clique of size r, which means that $S \cap c^+_{last}$ is empty and thus the first item is vacuously satisfied. Secondly, note that several cliques C might satisfy the two previous conditions. In that case, pick one such clique arbitrarily. This definition of C and c ensures that $S \cap c^+$ is an independent set in the set V_1 defined in the construction above (it might be empty, but we also consider this case). Thus, it will be picked in the second phase as some S_1 in $\mathcal{S}_1$ and for this choice, each time V_2 is considered, the fact that C is chosen to minimize its largest element c guarantees that there must be a vertex of S in the $(Ram(r,k)+1)$ largest vertices in V_2: either $c \in S$ and we are done, or S must intersect one of the $Ram(r,k)$ largest elements of $V_2 \setminus \{c\}$, otherwise there would be an an r-clique contradicting the choice of C. This shows that $S \in \mathcal{S}_C$, which concludes our proof. ◀

4 Positive results II

4.1 Key ingredient: Iterative expansion and Ramsey extraction

In this section, we present the main idea of our algorithms. It is a generalization of iterative expansion, which itself is the maximization version of the well-known iterative compression technique. Iterative compression is a useful tool for designing parameterized algorithms for subset problems (*i.e.* problems where a solution is a subset of some set of elements: vertices of a graph, variables of a logic formula...*etc.*) [6, 18]. Although it has been mainly used for minimization problems, iterative compression has been successfully applied for maximization problems as well, under the name *iterative expansion* [5]. Roughly speaking, when the problem consists in finding a solution of size at least k, the iterative expansion technique consists in solving the problem where a solution S of size $k-1$ is given in the input, in the hope that this solution will imply some structure in the instance. In the following, we consider an extension of this approach where, instead of a single smaller solution, one is given a set of $f(k)$ smaller solutions $S_1, \dots, S_{f(k)}$. As we will see later, we can further add more constraints on the sets $S_1, \dots, S_{f(k)}$. Notice that all the results presented in this sub-section (Lemmas 7 and 10 in particular) hold for any hereditary graph class (including the class of all graphs). The use of properties inherited from particular graphs (namely, H-free graphs in our case) will only appear in Sections 4.2 and 4.3.

▶ **Definition 6.** For a function $f : \mathbb{N} \to \mathbb{N}$, the f-Iterative Expansion MIS takes as input a graph G, an integer k, and a set of $f(k)$ independent sets $S_1, \dots, S_{f(k)}$, each of size $k-1$. The objective is to find an independent set of size k in G, or to decide that such an independent set does not exist.

▶ **Lemma 7.** *($\star$) Let $\mathcal{G}$ be a hereditary graph class.* MIS *is FPT in $\mathcal{G}$ iff f-*Iterative Expansion MIS *is FPT in $\mathcal{G}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.*

We will actually prove a stronger version of this result, by adding more constraints on the input sets $S_1, \dots, S_{f(k)}$, and show that solving the expansion version on this particular kind of input is enough to obtain the result for MIS.

▶ **Definition 8.** Given a graph G and a set of $k-1$ vertex-disjoint cliques of G, $\mathcal{C} = \{C_1, \dots, C_{k-1}\}$, each of size q, we say that $\mathcal{C}$ is a set of *Ramsey-extracted cliques of size* q if the conditions below hold. Let $C_r = \{c_j^r : j \in \{1, \dots, q\}\}$ for every $r \in \{1, \dots, k-1\}$.

- For every $j \in [q]$, the set $\{c_j^r : r \in \{1, \dots, k-1\}\}$ is an independent set of G of size $k-1$.
- For any $r \neq r' \in \{1, \dots, k-1\}$, one of the four following case can happen:
  (i) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \notin E(G)$
  (ii) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j \neq j'$
  (iii) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j < j'$
  (iv) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j > j'$
  In the case (i) (resp. (ii)), we say that the relation between C_r and $C_{r'}$ is *empty* (resp. *full*[5]). In case (iii) or (iv), we say the relation is *semi-full*.

Observe, in particular, that a set $\mathcal{C}$ of $k-1$ Ramsey-extracted cliques of size q can be partitionned into q independent sets of size $k-1$. As we will see later, these cliques will allow us to obtain more structure with the remaining vertices if the graph is H-free. Roughly speaking, if q is large, we will be able to extract from $\mathcal{C}$ another set $\mathcal{C}'$ of $k-1$ Ramsey-extracted cliques of size $q' < q$, such that every clique is a module[6] with respect to the solution $x_1^*, \dots, x_k^*$ we are looking for. Then, by guessing the structure of the adjacencies between $\mathcal{C}'$ and the solution, we will be able to identify from the remaining vertices k sets $X_1, \dots, X_k$, where each X_i has the same neighborhood as x_i^* *w.r.t.* $\mathcal{C}'$, and plays the role of "candidates" for this vertex. For a function $f : \mathbb{N} \to \mathbb{N}$, we define the following problem:

▶ **Definition 9.** The f-Ramsey-extracted Iterative Expansion MIS problem takes as input an integer k and a graph G whose vertices are partitionned into non-empty sets $X_1 \cup \dots \cup X_k \cup C_1 \cup \dots \cup C_{k-1}$, where:

- $\{C_1, \dots, C_{k-1}\}$ is a set of $k-1$ Ramsey-extracted cliques of size $f(k)$
- any independent set of size k in G is contained in $X_1 \cup \dots \cup X_k$
- $\forall i \in \{1, \dots, k\}$, $\forall v, w \in X_i$ and $\forall j \in \{1, \dots, k-1\}$, $N(v) \cap C_j = N(w) \cap C_j = \emptyset$ or $N(v) \cap C_j = N(w) \cap C_j = C_j$
- the following bipartite graph $\mathcal{B}$ is connected: $V(\mathcal{B}) = B_1 \cup B_2$, $B_1 = \{b_1^1, \dots, b_k^1\}$, $B_2 = \{b_1^2, \dots, b_{k-1}^2\}$ and $b_j^1 b_r^2 \in E(\mathcal{B})$ iff X_j and C_r are adjacent.

The objective is to find an independent set S in G of size at least k, or to decide that G does not contain an independent set S such that $S \cap X_i \neq \emptyset$ for all $i \in \{1, \dots, k\}$.

▶ **Lemma 10.** *Let $\mathcal{G}$ be a hereditary graph class. If there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that f-Ramsey-extracted Iterative Expansion MIS is FPT in $\mathcal{G}$, then g-Iterative Expansion MIS is FPT in $\mathcal{G}$, where $g(x) = Ram_{\ell}(f(x)2^{x(x-1)})$ $\forall x \in \mathbb{N}$, with $\ell_x = 2^{(x-1)^2}$.*

Proof. Let $f : \mathbb{N} \to \mathbb{N}$ be such a function, and let G, k and $\mathcal{S} = \{S_1, \dots, S_{g(k)}\}$ be an input of g-Iterative Expansion MIS. Recall that the objective is to find an independent set of size k in G, or to decide that such an independent set does not exist. If G contains an independent set of size k, then either there is one intersecting some set of $\mathcal{S}$, or every independent set of size k avoids the sets in $\mathcal{S}$. In order to capture the first case, we branch on every vertex v of the sets in $\mathcal{S}$, and make a recursive call with parameter $G \setminus N[v]$, $k-1$.

[5] Remark that in this case, the graph induced by $C_r \cup C_{r'}$ is the complement of a perfect matching.
[6] A set of vertices M is a module if every vertex $v \notin M$ is adjacent to either all vertices of M, or none.

In the remainder of the algorithm, we thus assume that any independent set of size k in G avoids every set of $\mathcal{S}$.

We choose an arbitrary ordering of the vertices of each S_j. Let us denote by s_j^r the r^{th} vertex of S_j. Notice that given an ordered pair of sets of $k-1$ vertices (A, B), there are $\ell_k = 2^{(k-1)^2}$ possible sets of edges between these two sets. Let us denote by $c_1, \ldots, c_{2^{(k-1)^2}}$ the possible sets of edges, called *types*. We define an auxiliary edge-colored graph H whose vertices are in one-to-one correspondence with $S_1, \ldots, S_{g(k)}$, and, for $i < j$, there is an edge between S_i and S_j of color γ iff the type of (S_i, S_j) is γ. By Ramsey's theorem, since H has $Ram_{\ell_k}(f(k)2^{k(k-1)})$ vertices, it must admit a monochromatic clique of size at least $h(k) = f(k)2^{k(k-1)}$. *W.l.o.g.*, the vertex set of this clique corresponds to $S_1, \ldots, S_{h(k)}$. For $p \in \{1, \ldots, k-1\}$, let $C_p = \{s_j^p, \ldots, s_{h(k)}^p\}$. Observe that the Ramsey extraction ensures that each C_p is either a clique or an independent set. If C_p is an independent set for some r, then we can immediately conclude, since $h(k) \geq k$. Hence, we suppose that C_p is a clique for every $p \in \{1, \ldots, k-1\}$. We now prove that $C_1, \ldots, C_{k-1}$ are Ramsey-extracted cliques of size $h(k)$. First, by construction, for every $j \in \{1, \ldots, h(k)\}$, the set $\{s_j^p : p = 1, \ldots, k-1\}$ is an independent set. Then, let c be the type of the clique obtained previously, represented by the adjacencies between two sets (A, B), each of size $k-1$. For every $p \in \{1, \ldots, k-1\}$, let a_p (resp. b_p) be the a^{th} vertex of A (resp. B). Let $p, q \in \{1, \ldots, t\}$, $p \neq q$. If any of $a_p b_q$ and $a_q b_p$ are edges in type c, then there is no edge between C_p and C_q, and their relation is thus empty. If both edges $a_p b_q$ and $a_q b_p$ exist in c, then the relation between C_p and C_q is full. Finally if exactly one edge among $a_p b_q$ and $a_q b_p$ exists in c, then the relation between C_p and C_q is semi-full. This concludes the fact that $\mathcal{C} = \{C_1, \ldots, C_{k-1}\}$ are Ramsey-extracted cliques of size $h(k)$.

Suppose that G has an independent set $X^* = \{x_1^*, \ldots, x_k^*\}$. Recall that we assumed previously that X^* is contained in $V(G) \setminus (C_1 \cup \cdots \cup C_{k-1})$. The next step of the algorithm consists in branching on every subset of $f(k)$ indices $J \subseteq \{1, \ldots, h(k)\}$, and restrict every set C_p to $\{s_j^p : j \in J\}$. For the sake of readability, we keep the notation C_p to denote $\{s_j^p : j \in J\}$ (the non-selected vertices are put back in the set of remaining vertices of the graph, *i.e.* we do not delete them). Since $h(k) = f(k)2^{k(k-1)}$, there must exist a branching where the chosen indices are such that for every $i \in \{1, \ldots, k\}$ and every $p \in \{1, \ldots, k-1\}$, x_i^* is either adjacent to all vertices of C_p or none of them. In the remainder, we may thus assume that such a branching has been made, with respect to the considered solution $X^* = \{x_1^*, \ldots, x_k^*\}$. Now, for every $v \in V(G) \setminus (C_1, \ldots, C_{k-1})$, if there exists $p \in \{1, \ldots, k-1\}$ such that $N(v) \cap C_p \neq \emptyset$ and $N(v) \cap C_p \neq C_p$, then we can remove this vertex, as we know that it cannot correspond to any x_i^*. Thus, we know that all the remaining vertices v are such that for every $p \in \{1, \ldots, k-1\}$, v is either adjacent to all vertices of C_p, or none of them.

In the following, we perform a color coding-based step on the remaining vertices. Informally, this color coding will allow us to identify, for every vertex x_i^* of the optimal solution, a set X_i of candidates, with the property that all vertices in X_i have the same neighborhood with respect to sets $C_1, \ldots, C_{k-1}$. We thus color uniformly at random the remaining vertices $V(G) \setminus (C_1, \ldots, C_{k-1})$ using k colors. The probability that the elements of X^* are colored with pairwise distinct colors is at least e^{-k}. We are thus reduced to the case of finding a *colorful*[7] independent set of size k. For every $i \in \{1, \ldots, k\}$, let X_i be the vertices of $V(G) \setminus (C_1, \ldots, C_{k-1})$ colored with color i. We now partition every set X_i into at most 2^{k-1} subsets $X_i^1, \ldots, X_i^{2^{k-1}}$, such that for every $j \in \{1, \ldots, 2^{k-1}\}$, all vertices of X_i^j have the same neighborhood with respect to the sets $C_1, \ldots, C_{k-1}$ (recall that every vertex of

[7] A set of vertices is called *colorful* if it is colored with pairwise distinct colors.

$V(G) \setminus (C_1, \ldots, C_{k-1})$ is adjacent to all vertices of C_p or none, for each $p \in \{1, \ldots, k-1\}$). We branch on every tuple $(j_1, \ldots, j_k) \in \{1, \ldots, 2^{k-1}\}$. Clearly the number of branchings is bounded by a function of k only and, moreover, one branching $(j_1, \ldots, j_k)$ is such that x_i^* has the same neighborhood in $C_1 \cup \cdots \cup C_{k-1}$ as vertices of $X_i^{j_i}$ for every $i \in \{1, \ldots, k\}$. We assume in the following that such a branching has been made. For every $i \in \{1, \ldots, k\}$, we can thus remove vertices of X_i^j for every $j \neq j_i$. For the sake of readability, we rename $X_i^{j_i}$ as X_i. Let $\mathcal{B}$ be the bipartite graph with vertex bipartition (B_1, B_2), $B_1 = \{b_1^1, \ldots, b_k^1\}$, $B_2 = \{b_1^2, \ldots, b_{k-1}^2\}$, and $b_i^1 b_p^2 \in E(\mathcal{B})$ iff x_i^* is adjacent to C_p. Since every x_i^* has the same neighborhood as X_i with respect to $C_1, \ldots, C_{k-1}$, this bipartite graph actually corresponds to the one described in Definition 9 representing the adjacencies between X_i's and C_p's. We now prove that it is connected. Suppose it is not. Then, since $|B_1| = k$ and $|B_2| = k-1$, there must be a component with as many vertices from B_1 as vertices from B_2. However, in this case, using the fixed solution X^* on one side and an independent set of size $k-1$ in $C_1 \cup \cdots \cup C_{k-1}$ on the other side, it implies that there is an independent set of size k intersecting $\cup_{p=1}^{k-1} C_p$, a contradiction.

Hence, all conditions of Definition 9 are now fulfilled. It now remains to find an independent set of size k disjoint from the sets $\mathcal{C}$, and having a non-empty intersection with X_i, for every $i \in \{1, \ldots, k\}$. We thus run an algorithm solving f-Ramsey-extracted Iterative Expansion MIS on this input, which concludes the algorithm. ◀

The proof of the following result is immediate, by using successively Lemmas 7 and 10.

▶ **Theorem 11.** *Let $\mathcal{G}$ be a hereditary graph class. If f-*Ramsey-extracted Iterative Expansion MIS *is FPT in $\mathcal{G}$ for some computable function f, then* MIS *is FPT in $\mathcal{G}$.*

We now apply this framework to two families of graphs H.

4.2 Clique minus a smaller clique

▶ **Theorem 12.** *($\star$) For any $r \geq 2$ and $s < r$,* MIS *in $(K_r \setminus K_s)$-free graphs is FPT if $s \leq 3$, and $W[1]$-hard otherwise.*

4.3 Clique minus a complete bipartite graph

For every three positive integers r, s_1, s_2 with $s_1 + s_2 < r$, we consider the graph $K_r \setminus K_{s_1,s_2}$. Another way to see $K_r \setminus K_{s_1,s_2}$ is as a P_3 *of cliques* of size s_1, $r - s_1 - s_2$, and s_2. More formally, every graph $K_r \setminus K_{s_1,s_2}$ can be obtained from a P_3 by adding $s_1 - 1$ false twins of the first vertex, $r - s_1 - s_2 - 1$, for the second, and $s_2 - 1$, for the third.

▶ **Theorem 13.** *$\forall r \geq 2$ and $s_1 \leq s_2$ s.t. $s_1 + s_2 < r$,* MIS *in $K_r \setminus K_{s_1,s_2}$-free graphs is FPT.*

Proof. It is more convenient to prove the result for $K_{3r} \setminus K_{r,r}$-free graphs, for any positive integer r. It implies the theorem by choosing this new r to be larger than s_1, s_2, and $r - s_1 - s_2$. We will show that for $f(x) := 3r$ for every $x \in \mathbb{N}$, f-Ramsey-extracted Iterative Expansion MIS in $K_{3r} \setminus K_{r,r}$-free graphs is FPT. By Theorem 11, this implies that MIS is FPT in this class. Let $C_1, \ldots, C_{k-1}$ (whose union is denoted by $\mathcal{C}$) be the Ramsey-extracted cliques of size $3r$, which can be partitionned, as in Definition 9, into $3r$ independent sets $S_1, \ldots, S_{3r}$, each of size $k-1$. Let $\mathcal{X} = \bigcup_{i=1}^{k} X_i$ be the set in which we are looking for an independent set of size k. We recall that between any X_i and any C_j there are either all the edges or none. Hence, the whole interaction between $\mathcal{X}$ and $\mathcal{C}$ can be described by the bipartite graph $\mathcal{B}$ described in Definition 9. Firstly, we can assume that each X_i is of size at least $Ram(r, k)$, otherwise we can branch on $Ram(r, k)$ choices to find one vertex in an optimum solution. By Ramsey's theorem, we can assume that each X_i contains a clique of size r (if it contains an independent set of size k, we are done). Our general strategy is to leverage the fact that the input graph is $(K_{3r} \setminus K_{r,r})$-free to describe the structure of $\mathcal{X}$. Hopefully, this structure will be sufficient to solve our problem in FPT time.

We define an auxiliary graph Y with $k-1$ vertices. The vertices $y_1, \dots, y_{k-1}$ of Y represent the Ramsey-extracted cliques of $\mathcal{C}$ and two vertices y_i and y_j are adjacent iff the relation between C_i and C_j is not empty (equivalently the relation is full or semi-full). It might seem peculiar that we concentrate the structure of $\mathcal{C}$, when we will eventually discard it from the graph. It is an indirect move: the simple structure of $\mathcal{C}$ will imply that the interaction between $\mathcal{X}$ and $\mathcal{C}$ is simple, which in turn, will severely restrict the subgraph induced by $\mathcal{X}$. More concretely, in the rest of the proof, we will (1) show that Y is a clique, (2) deduce that $\mathcal{B}$ is a complete bipartite graph, (3) conclude that $\mathcal{X}$ cannot contain an induced $K_r^2 = K_r \uplus K_r$ and run the algorithm of Theorem 5.

Suppose that there is $y_{i_1}y_{i_2}y_{i_3}$ an induced P_3 in Y, and consider C_{i_1}, C_{i_2}, C_{i_3} the corresponding Ramsey-extracted cliques. For $s < t \in [3r]$, let $C_i^{s \to t} := C_i \cap \bigcup_{s \leqslant j \leqslant t} S_j$. In other words, $C_i^{s \to t}$ contains the elements of C_i having indices between s and t. Since $|C_i| = 3r$, each C_i can be partitionned into three sets, of r elements each: $C_i^{1 \to r}$, $C_i^{r+1 \to 2r}$ and $C_i^{2r+1 \to 3r}$. Recall that the relation between C_{i_1} and C_{i_2} (resp. C_{i_2} and C_{i_3}) is either full or semi-full, while the relation between C_{i_1} and C_{i_3} is empty. This implies that at least one of the four following sets induces a graph isomorphic to $K_{3r} \setminus K_{r,r}$:

- $C_{i_1}^{1 \to r} \cup C_{i_2}^{r+1 \to 2r} \cup C_{i_3}^{1 \to r}$
- $C_{i_1}^{1 \to r} \cup C_{i_2}^{r+1 \to 2r} \cup C_{i_3}^{2r+1 \to 3r}$
- $C_{i_1}^{2r+1 \to 3r} \cup C_{i_2}^{r+1 \to 2r} \cup C_{i_3}^{1 \to r}$
- $C_{i_1}^{2r+1 \to 3r} \cup C_{i_2}^{r+1 \to 2r} \cup C_{i_3}^{2r+1 \to 3r}$

Hence, Y is a disjoint union of cliques. Let us assume that Y is the union of at least two (maximal) cliques.

Recall that the bipartite graph $\mathcal{B}$ is connected. Thus there is $b_h^1 \in B_1$ (corresponding to X_h) adjacent to $b_i^2 \in B_2$ and $b_j^2 \in B_2$ (corresponding to C_i and C_j, respectively), such that y_i and y_j lie in two different connected components of Y (in particular, the relation between C_i and C_j is empty). Recall that X_h contains a clique of size at least r. This clique induces, together with any r vertices in C_i and any r vertices in C_j, a graph isomorphic to $K_{3r} \setminus K_{r,r}$; a contradiction. Hence, Y is a clique.

Now, we can show that $\mathcal{B}$ is a complete bipartite graph. Each X_h has to be adjacent to at least one C_i (otherwise this trivially contradicts the connectedness of $\mathcal{B}$). If X_h is not linked to C_j for some $j \in \{1, \dots, k-1\}$, then a clique of size r in X_h (which always exists) induces, together with $C_i^{1 \to r} \cup C_j^{2r+1 \to 3r}$ or with $C_i^{2r+1 \to 3r} \cup C_j^{1 \to r}$, a graph isomorphic to $K_{3r} \setminus K_{r,r}$.

Since $\mathcal{B}$ is a complete bipartite graph, every vertex of C_1 dominates all vertices of $\mathcal{X}$ In particular, $\mathcal{X}$ is in the intersection of the neighborhood of the vertices of some clique of size r. This implies that the subgraph induced by $\mathcal{X}$ is $(K_r \uplus K_r)$-free. Hence, we can run the FPT algorithm of Theorem 5 on this graph. ◀

5 Polynomial (Turing) kernels

In this section we investigate some special cases of Section 4.3, in particular when H is a clique of size r minus a claw with s branches, for $s < r$. Although Theorem 13 proves that MIS is FPT for every possible values of r and s, we show that when $s \geq r-2$, the problem admits a polynomial Turing kernel, while for $s \leq 2$, it admits a polynomial kernel. Notice that the latter result is somehow tight, as Corollary 18 shows that MIS cannot admit a polynomial kernel in $(K_r \setminus K_{1,s})$-free graphs whenever $s \geq 3$.

▶ **Theorem 14.** *(⋆) $\forall r \geq 2$, MIS in $(K_r \setminus K_{1,r-2})$-free graphs has a polynomial Turing kernel.*

▶ **Theorem 15.** *(⋆) $\forall r \geq 3$, MIS in $(K_r \setminus K_{1,2})$-free graphs has a kernel with $O(k^{r-1})$ vertices.*

Observe that a $(K_r \setminus K_2)$-free graph is $(K_{r+1} \setminus K_{1,2})$-free, hence, thus the previous result also applies to $(K_r \setminus K_2)$-free graphs, which answers a question of [8].

We now focus on kernel lower bounds.

▶ **Definition 16.** Given the graphs H, $H_1, \ldots, H_p$, we say that $(H_1, \ldots, H_p)$ is a multipartite decomposition of H if H is isomorphic to $H_1 + \cdots + H_p$. We say that $(H_1, \ldots, H_p)$ is maximal if, for every multipartite decomposition $(H'_1, \ldots, H'_q)$ of H, we have $p > q$.

It can easily be seen that for every graph H, a maximal multipartite decomposition of H is unique. We have the following:

▶ **Theorem 17.** *(⋆) Let H be any fixed graph, and let $H = H_1 + \cdots + H_p$ be the maximal multipartite decomposition of H. If, for some $i \in [p]$,* MIS *is NP-hard in H_i-free graphs, then* MIS *does not admit a polynomial kernel in H-free graphs unless NP $\subseteq$ coNP/poly.*

The next results shows that the polynomial kernel obtained in the previous section for $(K_r \setminus K_{1,s})$-free graphs, $s \leq 2$, is somehow tight.

▶ **Corollary 18.** *(⋆) For $r \geq 4$, and every $3 \leq s \leq r-1$,* MIS *in $(K_r \setminus K_{1,s})$-free graphs does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly.*

6 Conclusion and open problems

We started to unravel the FPT/$W[1]$-hard dichotomy for MIS in H-free graphs, for a fixed graph H. At the cost of one reduction, we showed that it is $W[1]$-hard as soon as H is not chordal, even if we simultaneously forbid induced $K_{1,4}$ and trees with at least two branching vertices. Tuning this construction, it is also possible to show that if a connected H is not roughly a "path of cliques" or a "subdivided claw of cliques", then MIS is $W[1]$-hard.

An interesting open problem is the case when H is the *cricket*, that is a triangle with two pending vertices, each attached to a different vertex

For disconnected graphs H, we obtained an FPT algorithm when H is a cluster (*i.e.*, a disjoint union of cliques). We conjecture that, more generally, the disjoint union of two easy cases is an easy case; formally, *if* MIS *is FPT in G-free graphs and in H-free graphs, then it is FPT in $G \uplus H$-free graphs.* A more anecdotal conclusion is the fact that the parameterized complexity of the problem on H-free graphs is now complete for every graph H on four vertices, including concerning the polynomial kernel question, whereas the $FPT/W[1]$-hard question remains open for only five graphs H on five vertices.

References

1 V. Alekseev. The Effect of Local Constraints on the Complexity of Determination of the Graph Independence Number. *Combinatorial-Algebraic Methods in Applied Mathematics*, pages 3–13, 1982. in Russian.

2 V. E. Alekseev. On the number of maximal independent sets in graphs from hereditary classes. *Combinatorial-Algebraic Methods in Discrete Optimization*, pages 5–8, 1991. (In Russian).

3 G. Bacsó, D. Lokshtanov, D. Marx, M. Pilipczuk, Z. Tuza, and E. Jan van Leeuwen. Subexponential-time Algorithms for Maximum Independent Set in P_t-free and Broom-free Graphs. *CoRR*, abs/1804.04077, 2018.

4 É. Bonnet, N. Bousquet, P. Charbit, S. Thomassé, and R. Watrigant. Parameterized Complexity of Independent Set in H-Free Graphs. *CoRR*, 2018. `arXiv:1810.04620`.

5 J. Chen, Y. L., S. Lu, S. S., and F. Zhang. Iterative Expansion and Color Coding: An Improved Algorithm for 3D-Matching. *ACM Trans. Algorithms*, 8(1):6:1–6:22, 2012.

6 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

7 K. Dabrowski. *Structural Solutions to Maximum Independent Set and Related Problems*. PhD thesis, University of Warwick, 2012.

8 K. Dabrowski, V. V. Lozin, H. Müller, and D. Rautenbach. Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number. *J. Discrete Algorithms*, 14:207–213, 2012.

9 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

10 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

11 H. Perret du Cray and I. Sau. Improved FPT algorithms for weighted independent set in bull-free graphs. *Discrete Mathematics*, 341(2):451–462, 2018.

12 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

13 A. Grzesik, T. Klimosova, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for Maximum Weight Independent Set on P_6-free graphs. *CoRR*, abs/1707.05491, 2017.

14 D. Hermelin, M. Mnich, and E. J. van Leeuwen. Parameterized Complexity of Induced Graph Matching on Claw-Free Graphs. *Algorithmica*, 70(3):513–560, 2014.

15 T. Karthick. Independent Sets in Some Classes of $\mathrm{S}_{i,j,k}$-free Graphs. *J. Comb. Optim.*, 34(2):612–630, August 2017.

16 T. Karthick and F. Maffray. Maximum weight independent sets in classes related to claw-free graphs. *Discrete Applied Mathematics*, 216:233–239, 2017.

17 D. Lokshtanov, M. Vatshelle, and Y. Villanger. Independent Set in P_5-Free Graphs in Polynomial Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 570–581, 2014.

18 B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

19 S. Thomassé, N. Trotignon, and K. Vuskovic. A Polynomial Turing-Kernel for Weighted Independent Set in Bull-Free Graphs. *Algorithmica*, 77(3):619–641, 2017.

20 D. Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.