



HAL
open science

Parameterized Complexity of Independent Set in H-free Graphs

Edouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphane Thomassé, Rémi Watrigant

► **To cite this version:**

Edouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphane Thomassé, Rémi Watrigant. Parameterized Complexity of Independent Set in H-free Graphs. *Algorithmica*, 2020, Parameterized and Exact Computation, IPEC 2018, 82 (8), pp.2360-2394. 10.1007/s00453-020-00730-6 . hal-03015353v2

HAL Id: hal-03015353

<https://hal.science/hal-03015353v2>

Submitted on 19 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 **Parameterized Complexity of Independent Set in**
2 **H-Free Graphs**

3 **Édouard Bonnet · Nicolas Bousquet ·**
4 **Pierre Charbit · Stéphan Thomassé ·**
5 **Rémi Watrigant**

6
7 Received: date / Accepted: date

8 **Abstract** In this paper, we investigate the complexity of MAXIMUM INDE-
9 PENDENT SET (MIS) in the class of H -free graphs, that is, graphs excluding
10 a fixed graph as an induced subgraph. Given that the problem remains NP -
11 hard for most graphs H , we study its fixed-parameter tractability and make
12 progress towards a dichotomy between FPT and $W[1]$ -hard cases. We first
13 show that MIS remains $W[1]$ -hard in graphs forbidding simultaneously $K_{1,4}$,
14 any finite set of cycles of length at least 4, and any finite set of trees with at
15 least two branching vertices. In particular, this answers an open question of
16 Dabrowski *et al.* concerning C_4 -free graphs. Then we extend the polynomial
17 algorithm of Alekseev when H is a disjoint union of edges to an FPT algo-
18 rithm when H is a disjoint union of cliques. We also provide a framework for

A preliminary version of this paper appeared in the proceedings of the 13th International Symposium on Parameterized and Exact Computation (IPEC 2018).

Édouard Bonnet
Université de Lyon, CNRS, ENS Lyon, Université Claude Bernard Lyon 1, LIP UMR5668,
France
E-mail: edouard.bonnet@ens-lyon.fr

Nicolas Bousquet
CNRS, G-Scop Laboratory, Grenoble-INP, France
E-mail: nicolas.bousquet@grenoble-inp.fr

Pierre Charbit
Université Paris Diderot, IRIF, France
E-mail: charbit@irif.fr

Stéphan Thomassé
Université de Lyon, CNRS, ENS Lyon, Université Claude Bernard Lyon 1, LIP UMR5668,
France
E-mail: stephan.thomasse@ens-lyon.fr

Rémi Watrigant
Université de Lyon, CNRS, ENS Lyon, Université Claude Bernard Lyon 1, LIP UMR5668,
France
E-mail: remi.watrigant@ens-lyon.fr

19 solving several other cases, which is a generalization of the concept of *iterative*
 20 *expansion* accompanied by the extraction of a particular structure using Ram-
 21 sey’s theorem. Iterative expansion is a maximization version of the so-called
 22 *iterative compression*. We believe that our framework can be of independent
 23 interest for solving other similar graph problems. Finally, we present positive
 24 and negative results on the existence of polynomial (Turing) kernels for several
 25 graphs H .

26 **Keywords** Parameterized Algorithms · Independent Set · H-Free Graphs

27 1 Introduction

28 Given a simple graph G , a set of vertices $S \subseteq V(G)$ is an *independent set* if
 29 the vertices of this set are all pairwise non-adjacent. Finding an independent
 30 set with maximum cardinality is a fundamental problem in algorithmic graph
 31 theory, and is known as the MIS problem (MIS, for short) [15]. In general
 32 graphs, it is not only NP -hard, but also not approximable within $O(n^{1-\epsilon})$ for
 33 any $\epsilon > 0$ unless $P = NP$ [28], and $W[1]$ -hard parameterized by the solution
 34 size [14] (unless otherwise stated, n always denotes the number of vertices of
 35 the input graph). Thus, it seems natural to study the complexity of MIS in
 36 restricted graph classes. One natural way to obtain such a restricted graph
 37 class is to forbid some given pattern to appear in the input. For a fixed graph
 38 H , we say that a graph is *H-free* if it does not contain H as an induced
 39 subgraph. Unfortunately, it turns out that for most graphs H , MIS in H -
 40 free graphs remains NP -hard, as shown by a very simple reduction observed
 41 independently by Poljak [24] and Alekseev [1]:

42 **Theorem 1 ([1, 24])** *Let H be a connected graph which is neither a path nor*
 43 *a subdivision of the claw. Then MIS is NP -hard in H -free graphs.*

44 On the positive side, the case of P_t -free graphs has attracted a lot of atten-
 45 tion during the last decade. While it is still open whether there exists $t \in \mathbb{N}$
 46 for which MIS is NP -hard in P_t -free graphs, quite involved polynomial-time
 47 algorithms were discovered for P_5 -free graphs [20], and very recently for P_6 -
 48 free graphs [16]. In addition, we can also mention the recent following result:
 49 MIS admits a subexponential algorithm running in time $2^{O(\sqrt{tn \log n})}$ in P_t -free
 50 graphs for every $t \in \mathbb{N}$ [3].

51 The second open question concerns subdivisions of the claw. Let $S_{i,j,k}$ be a
 52 tree with exactly three vertices of degree one, being at distance i , j and k from
 53 the unique vertex of degree three. The complexity of MIS is still open in $S_{1,2,2}$ -
 54 free graphs and $S_{1,1,3}$ -free graphs. In this direction, the only positive results
 55 concern some subclasses: it is polynomial-time solvable in $(S_{1,2,2}, S_{1,1,3}, \textit{dart})$ -
 56 free graphs [18], $(S_{1,1,3}, \textit{banner})$ -free graphs and $(S_{1,1,3}, \textit{bull})$ -free graphs [19],
 57 where *dart*, *banner* and *bull* are particular graphs on five vertices.

58 Given the large number of graphs H for which the problem remains NP -
 59 hard, it seems natural to investigate the existence of fixed-parameter tractable

(FPT) algorithms¹, that is, determining the existence of an independent set of size k in a graph with n vertices in time $f(k)n^c$ 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.* [12] (see also Dabrowski's PhD manuscript [11]) 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 [17]. 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 [27] then improved [9].

Finally, a subset of this paper's authors recently settled several other cases [5], such as the *cricket* graph, the \bar{P} graph, or the path of size four where all but one endpoint are replaced by a clique of fixed size.

1.1 Our results

In Section 2, we present three reductions proving $W[1]$ -hardness of MIS in graphs 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. We actually show the stronger result that MIS remains $W[1]$ -hard in graphs simultaneously excluding these graphs as induced subgraphs. We propose a definition of a graph decomposition whose aim is to capture all graphs which can be excluded using our reductions.

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, when H is a clique minus a triangle, and when H is the so-called *gem* graph.

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.

100 Finally, in Section 5, we focus on the existence of polynomial (Turing)
 101 kernels. We first strengthen some results of the previous section by providing
 102 polynomial (Turing) kernels in the case where H is a clique minus a claw.
 103 Then, we prove that for many H , MIS on H -free graphs does not admit a
 104 polynomial kernel, unless $NP \subseteq coNP/poly$.

105 Our results allow to obtain the complete trichotomy polynomial/polynomial
 106 kernel (PK)/no PK but polynomial Turing kernel/ $W[1]$ -hard for all possible
 107 graphs on four vertices.

108 1.2 Notation

109 For classical notation related to graph theory or fixed-parameter tractable
 110 algorithms, we refer the reader to the monographs [13] and [14], respectively.
 111 For an integer $r \geq 2$ and a graph H with vertex set $V(H) = \{v_1, \dots, v_{n_H}\}$
 112 with $n_H \leq r$, we denote by $K_r \setminus H$ the graph with vertex set $\{1, \dots, r\}$ and
 113 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
 114 write $G \setminus X$ to denote $G[V(G) \setminus X]$. For two graphs G and H , we denote
 115 by $G \uplus H$ the *disjoint union* operation, that is, the graph with vertex set
 116 $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. We denote by $G + H$ the *join*
 117 operation of G and H , that is, the graph with vertex set $V(G) \cup V(H)$ and
 118 edge set $E(G) \cup E(H) \cup \{uv : u \in V(G), v \in V(H)\}$. For two integers r, k ,
 119 we denote by $Ram(r, k)$ the Ramsey number of r and k , *i.e.* the number such
 120 that every graph with at least $Ram(r, k)$ vertices contains either a clique of
 121 size r or an independent set of size k . We write for short $Ram(k) = Ram(k, k)$.
 122 Finally, for $\ell, k > 0$, we denote by $Ram_\ell(k)$ the minimum order of a complete
 123 graph whose edges are colored with ℓ colors to contain a monochromatic clique
 124 of size k . The following bounds are known: $Ram(r, k) \leq \binom{r+k-2}{r-1} = \binom{r+k-2}{k-1}$,
 125 and $Ram_\ell(k) \leq k^{\ell k}$.

126 2 $W[1]$ -hardness

127 2.1 Main reduction

128 We show the following:

129 **Theorem 2** *For any $p_1 \geq 4$ and $p_2 \geq 1$, MIS remains $W[1]$ -hard in graphs*
 130 *excluding simultaneously the following graphs as induced subgraphs:*

- 131 – $K_{1,4}$
- 132 – C_4, \dots, C_{p_1}
- 133 – any tree T with two branching vertices² at distance at most p_2 .

134 *Proof.* Let $p = \max\{p_1, p_2\}$. We reduce from GRID TILING, where the input
 135 is composed of k^2 sets $S_{i,j} \subseteq [m] \times [m]$ ($0 \leq i, j \leq k-1$), called *tiles*, each

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

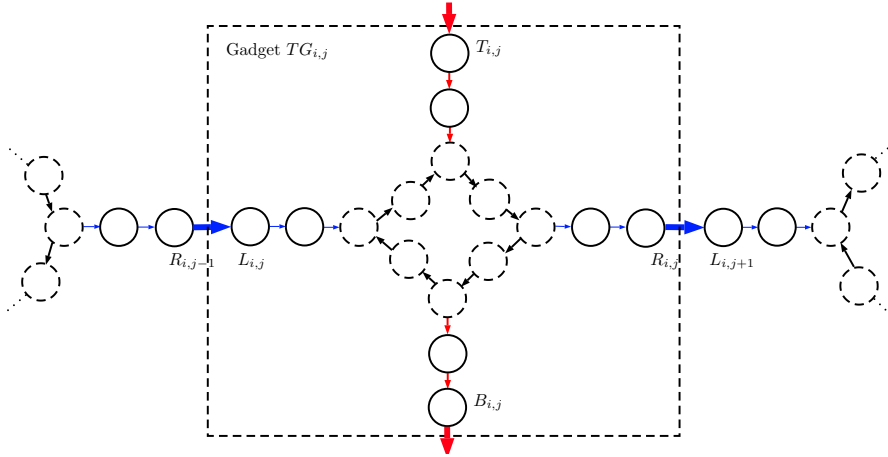


Fig. 1 Gadget $TG_{i,j}$ representing a tile and its adjacencies with $TG_{i,j-1}$ and $TG_{i,j+1}$, for $p = 1$. Each circle is a main clique on n vertices: dashed cliques are the cycle cliques (those of them connected to three other cliques are branching cliques), while others are path cliques. Black, blue and red arrows represent respectively type T_h , T_r and T_c edges (bold arrows are between two gadgets). Figures 2 and 3 represent some adjacencies in more details.

136 composed of n elements. The objective of GRID TILING is to find an element
 137 $s_{i,j}^* \in S_{i,j}$ for each $0 \leq i, j \leq k-1$, such that $s_{i,j}^*$ agrees in the first coordinate
 138 with $s_{i,j+1}^*$, and agrees in the second coordinate with $s_{i+1,j}^*$, for every $0 \leq$
 139 $i, j \leq k-1$ (here and henceforth, $i+1$ and $j+1$ are taken modulo k). In such
 140 case, we say that $\{s_{i,j}^*, 0 \leq i, j \leq k-1\}$ is a *feasible solution* of the instance.
 141 It is known that GRID TILING is $W[1]$ -hard parameterized by k [10, 21].

142 Before describing formally the reduction, let us give some definitions and
 143 ideas. Given $s = (a, b)$ and $s' = (a', b')$, we say that s is *row-compatible* (resp.
 144 *column-compatible*) with s' if $a \geq a'$ (resp. $b \geq b'$)³. Observe that a solution
 145 $\{s_{i,j}^*, 0 \leq i, j \leq k-1\}$ is feasible if and only if $s_{i,j}^*$ is row-compatible with
 146 $s_{i,j+1}^*$ and column-compatible with $s_{i+1,j}^*$ for every $0 \leq i, j \leq k-1$.

147 We will represent each tile by a gadget partitioned into a constant number
 148 of cliques of size n . The vertices of each clique are in one-to-one correspondence
 149 with the elements of the corresponding tile. Overall the cliques will be arranged
 150 in a grid-like structure with degree three. By that we mean that a clique will be
 151 linked to at most three other cliques. While most of the cliques will have only
 152 two neighboring cliques, a clique linked to three other cliques will be called
 153 *branching clique*. The row-compatibility (resp. column-compatibility) will be
 154 encoded with a relatively simple interaction between two adjacent cliques. The
 155 main difficulty will be to prevent the undesired induced subgraphs to appear
 156 in the vicinity of branching cliques. We now formally describe the reduction.

³ Notice that the row-compatibility (resp. column-compatibility) relation is not symmetric.

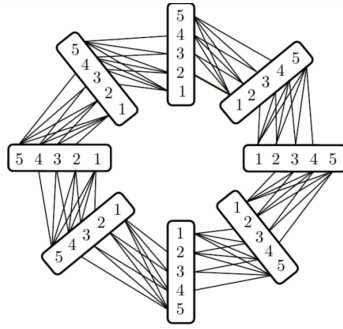


Fig. 2 Adjacencies between cycle cliques (represented by dashed circles in Figure 1).

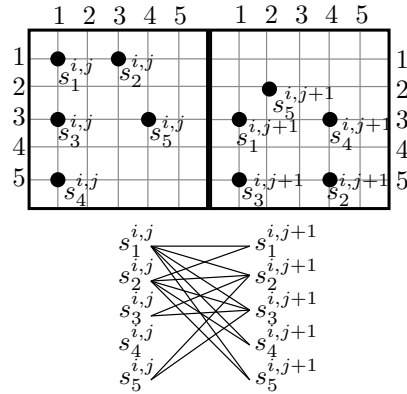


Fig. 3 Two consecutive tiles and the representation of their adjacencies (representing type T_r adjacencies).

157 2.1.1 Tile gadget.

158 For every tile $S_{i,j} = \{s_1^{i,j}, \dots, s_n^{i,j}\}$, we construct a *tile gadget* $TG_{i,j}$, depicted
 159 in Figure 1. Notice that this gadget shares some ideas with the $W[1]$ -hardness
 160 proof for MIS in $K_{1,4}$ -free graphs by Hermelin *et al.* [17]. To define this gadget,
 161 we first describe an oriented graph with three types of arcs (type T_h , T_r and
 162 T_c , which respectively stands for *half-graph*, *row* and *column*, and this naming
 163 will become clearer later), and then explain how to represent the vertices and
 164 arcs of this graph to get the concrete gadget. Consider first a directed cycle on
 165 $4p + 4$ vertices c_1, \dots, c_{4p+4} with arcs of type T_h . Then consider four oriented
 166 paths on $p + 1$ vertices: P_1, P_2, P_3 and P_4 . P_1 and P_3 are composed of arcs of
 167 type T_c , while P_2 and P_4 are composed of arcs of type T_r . Put an arc of type
 168 T_c between:

- 169 – the last vertex of P_1 and c_1 ,
- 170 – c_{2p+3} and the first vertex of P_3 ,

171 and an arc of type T_r between:

- 172 – c_{p+2} and the first vertex of P_2 ,
 173 – the last vertex of P_4 and c_{3p+4} .

174 Now, we replace every vertex of this oriented graph by a clique on n vertices,
 175 and fix an arbitrary ordering on the vertices of each clique. The i^{th} vertex
 176 in this ordering is said to *have index i* . For each arc of type T_h between c
 177 and c' , add a half-graph⁴ between the corresponding cliques: connect the a^{th}
 178 vertex of the clique representing c with the b^{th} vertex of the clique representing
 179 c' whenever $a > b$. For every arc of type T_r from a vertex c to a vertex c' ,
 180 connect the a^{th} vertex of the clique representing c with the b^{th} vertex of the
 181 clique representing c' iff $s_a^{i,j}$ is *not* row-compatible with $s_b^{i,j}$. Similarly, for
 182 every arc of type T_c from a vertex c to a vertex c' , connect the a^{th} vertex of
 183 the clique representing C with the b^{th} vertex of the clique representing c' iff
 184 $s_a^{i,j}$ is *not* column-compatible with $s_b^{i,j}$.

185 The cliques corresponding to vertices of this gadget are called the *main*
 186 *cliques* of $TG_{i,j}$, and the cliques corresponding to the central cycle on $4p + 4$
 187 vertices are called the *cycle cliques*. The main cliques which are not cycle
 188 cliques are called *path cliques*. The cycle cliques adjacent to one path clique
 189 are called *branching cliques*. We call *cycle of cliques* the set of all cycle cliques
 190 present in the same gadget $TG_{i,j}$. Two cycle of cliques are said *consecutive*
 191 if they lie on two gadgets $TG_{i,j}$ and $TG_{i,j+1}$, or $TG_{i,j}$ and $TG_{i+1,j}$. A *path*
 192 *of cliques* is any subgraph induced by the cliques corresponding to vertices
 193 forming a directed path in the oriented preliminary graph.

194 Finally, the clique corresponding to the vertex of degree one in the path at-
 195 tached to c_1 (resp. c_{p+2} , c_{2p+3} , c_{3p+4}) is called the *top* (resp. *right*, *bottom*, *left*)
 196 clique of $TG_{i,j}$, denoted by $T_{i,j}$ (resp. $R_{i,j}$, $B_{i,j}$, $L_{i,j}$). Let $T_{i,j} = \{t_1^{i,j}, \dots, t_n^{i,j}\}$,
 197 $R_{i,j} = \{r_1^{i,j}, \dots, r_n^{i,j}\}$, $B_{i,j} = \{b_1^{i,j}, \dots, b_n^{i,j}\}$, and $L_{i,j} = \{\ell_1^{i,j}, \dots, \ell_n^{i,j}\}$. For the
 198 sake of readability, we might omit the superscripts i, j when it is clear from
 199 the context.

200 **Lemma 1** *Let K be an independent set of size $8(p + 1)$ in $TG_{i,j}$. Then:*

- 201 (a) K intersects all the cycle cliques on the same index $x \in [n]$;
 202 (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 $K \cap L_{i,j} = \{\ell_{x_\ell}\}$.

203 *Then:*

- 204 – $s_{x_\ell}^{i,j}$ is row-compatible with $s_x^{i,j}$ which is row-compatible with $s_{x_r}^{i,j}$, and
 205 – $s_{x_t}^{i,j}$ is column-compatible with $s_x^{i,j}$ which is column-compatible with $s_{x_b}^{i,j}$.

206 *Proof.* Observe that the vertices of $TG_{i,j}$ can be partitioned into $8(p + 1)$
 207 cliques (the main cliques), hence an independent set of size $8(p + 1)$ intersects
 208 each main clique on exactly one vertex. Let C_1 , C_2 and C_3 be three consecutive
 209 cycle cliques, and suppose K intersects C_1 (resp. C_2 , C_3) on the x_1^{th} (resp. x_2^{th} ,
 210 x_3^{th}) index. By definition of the gadget, it implies $x_1 \leq x_2 \leq x_3$. By applying

⁴ Notice that our definition of half-graph slightly 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).

211 the same argument from C_3 along the cycle, we obtain $x_3 \leq x_1$, which proves
 212 (a). The proof of (b) directly comes from the definition of the adjacencies
 213 between cliques of type T_r and T_c , and from the fact that K intersects all
 214 cycle cliques on the same index. \square

215 2.1.2 Attaching gadgets together.

216 For $i, j \in \{0, \dots, k-1\}$, we connect the right clique of $TG_{i,j}$ with the left
 217 clique of $TG_{i,j+1}$ in a “type T_r spirit”: for every $x, y \in [n]$, connect $r_x^{i,j} \in R_{i,j}$
 218 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
 219 connect the bottom clique of $TG_{i,j}$ with the top clique of $TG_{i+1,j}$ in a “type
 220 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 $s_x^{i,j}$
 221 is *not* column-compatible with $s_y^{i+1,j}$ (all incrementations of i and j are done
 222 modulo k). This terminates the construction of the graph G .

223 2.1.3 Equivalence of solutions.

224 We now prove that the input instance of GRID TILING is positive if and only
 225 if G has an independent set of size $k' = 8(p+1)k^2$. First observe that G has k^2
 226 tile gadgets, each composed of $8(p+1)$ main cliques, hence any independent
 227 set of size k' intersects each main clique on exactly one vertex. By Lemma 1,
 228 for all $i, j \in \{0, \dots, k-1\}$, K intersects the cycle cliques of $TG_{i,j}$ on the
 229 same index $x_{i,j}$. Moreover, if $K \cap R_{i,j} = \{r_x^{i,j}\}$ and $K \cap L_{i,j+1} = \{\ell_{x'}^{i,j+1}\}$,
 230 then, by construction of G , $s_x^{i,j}$ is row-compatible with $s_{x'}^{i,j+1}$. Similarly, if
 231 $K \cap B_{i,j} = \{b_x^{i,j}\}$ and $K \cap T_{i+1,j} = \{t_{x'}^{i+1,j}\}$, then, by construction of G , $s_x^{i,j}$
 232 is column-compatible with $s_{x'}^{i+1,j}$. By Lemma 1, it implies that $s_{x_{i,j}}^{i,j}$ is row-
 233 compatible with $s_{x_{i,j+1}}^{i,j+1}$ and column-compatible with $s_{x_{i+1,j}}^{i+1,j}$ (incrementations
 234 of i and j are done modulo k), thus $\{x_{i,j}^{i,j} : 0 \leq i, j \leq k-1\}$ is a feasible
 235 solution. Using similar ideas, one can prove that a feasible solution of the grid
 236 tiling instance implies an independent set of size k' in G .

237 2.1.4 Structure of the obtained graph.

238 Let us now prove that G does not contain the graphs mentioned in the state-
 239 ment as an induced subgraph:

240 **No $\mathbf{K}_{1,4}$.** We first prove that for every $0 \leq i, j \leq k-1$, the graph induced
 241 by the cycle cliques of $TG_{i,j}$ is claw-free. For the sake of contradiction, suppose
 242 that there exist three consecutive cycle cliques A, B and C containing a claw.
 243 W.l.o.g. we may assume that $b_x \in B$ is the center of the claw, and $a_\alpha \in A$,
 244 $b_\beta \in B$ and $c_\gamma \in C$ are the three endpoints. By construction of the gadgets
 245 (there is a half-graph between A and B and between B and C), we must have
 246 $\alpha < x < \gamma$. Now, observe that if $x < \beta$ then a_α must be adjacent to b_β , and
 247 if $\beta < x$, then b_β must be adjacent to c_γ , but both case are impossible since
 248 $\{a_\alpha, b_\beta, c_\gamma\}$ is supposed to be an independent set.

249 Similarly, each subgraph induced by P , a path of size $2(p+1)$ of cliques
 250 linking two consecutive cycles of cliques, is claw-free. Hence, for $K_{1,4}$ to appear
 251 in G its center would have to lie in a branching clique. However, in that case,
 252 a claw must exist either in the cycle of cliques or in P , which we already ruled
 253 out.

254 **No C_4, \dots, C_{p_1} .** The main argument is that the graph induced by any
 255 two main cliques does not contain any of these cycles. Then, we show that
 256 such a cycle cannot lie entirely in the cycle cliques of a single gadget $TG_{i,j}$.
 257 Indeed, if this cycle uses at most one vertex per main clique, then it must
 258 be of length at least $4p+4$. If it intersects a clique C on two vertices, then
 259 either it also intersect all the cycle cliques of the gadget, in which case it is of
 260 length $4p+5$, or it intersects an adjacent clique of C on two vertices, in which
 261 case these two cliques induce a C_4 , which is impossible. Similarly, such a cycle
 262 cannot lie entirely in a path between the main cliques of two gadgets. Finally,
 263 the main cliques of two gadgets are at distance at least $2(p+1)$, hence such a
 264 cycle cannot intersect the main cliques of two gadgets.

265 **No tree T with two branching vertices at distance at most p_2 .**
 266 Using the same argument as for the $K_{1,4}$ case, observe that the claws con-
 267 tained in G can only appear in the cycle cliques where the paths are attached.
 268 However, observe that these cliques are at distance $2(p+1) > p_2$, thus, such
 269 a tree T cannot appear in G . \square

270 As a direct consequence of Theorem 2, we get the following by setting
 271 $p_1 = p_2 = |V(H)| + 1$:

272 **Corollary 1** *If H is not chordal, or contains as an induced subgraph a $K_{1,4}$*
 273 *or a tree with two branching vertices, then MIS in H -free graphs is $W[1]$ -hard.*

274 2.2 Capturing Hard Graphs

275 We introduce two variants of the hardness construction of Theorem 2, which
 276 we refer to as the *first construction*. The *second construction* is obtained by
 277 replacing each interaction between two main cliques by an anti-matching, ex-
 278 cept the one interaction in the middle of the path cliques which remains a
 279 half-graph (see Figure 4, middle). In an anti-matching, the same elements in
 280 the two adjacent cliques define the only non-edges. The correctness of this
 281 new reduction is simpler since the propagation of a choice is now straightfor-
 282 ward. Observe however that the graph C_4 appears in this new construction.
 283 For the *third construction*, we start from the second construction and just add
 284 an anti-matching between two neighbors of each branching clique among the
 285 cycle cliques (see Figure 4, right). This anti-matching only constrains more the
 286 instance but does not destroy the intended solutions; hence the correctness.

287 To describe those connected graphs H which escape the disjunction of
 288 Theorem 2 (for which there is still a hope that MIS is FPT), we define a
 289 decomposition into cliques, similar yet different from clique graphs or tree
 290 decompositions of chordal graphs (a.k.a k -trees).

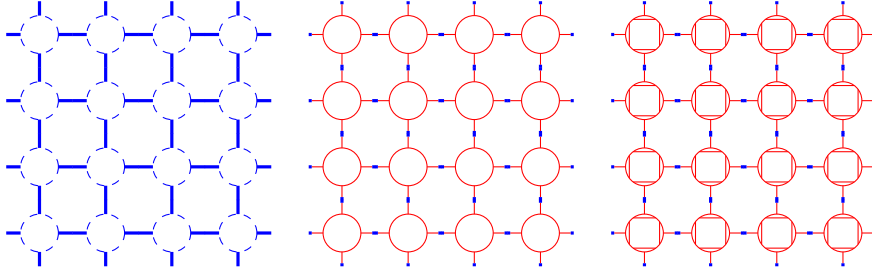


Fig. 4 A symbolic representation of the hardness constructions. To the left, only half-graphs (blue) are used between the cliques, as in the proof of Theorem 2. In the middle and to the right, the half-graphs (blue) are only used once in the middle of each path of cliques, and the rest of the interactions between the cliques are anti-matchings (red). The third construction (right) is a slight variation of the second (middle) where for each branching clique, we link by an anti-matching its two neighbors among the cycle cliques.

291 **Definition 1** Let T be a graph on ℓ vertices t_1, \dots, t_ℓ . We say that T is a
 292 *clique decomposition of H* if there is a partition of $V(H)$ into $(C_1, C_2, \dots, C_\ell)$
 293 such that:

- 294 – for each $i \in [\ell]$, $H[C_i]$ is a clique, and
- 295 – for each pair $i \neq j \in [\ell]$, if $H[C_i \cup C_j]$ is connected, then $t_i t_j \in E(T)$.

296 Observe that, in the above definition, we do not require T to be a tree.
 297 Two cliques C_i and C_j are said *adjacent* if $H[C_i \cup C_j]$ is connected. We also
 298 write a *clique decomposition on T (of H)* to denote the choice of an actual
 299 partition $(C_1, C_2, \dots, C_\ell)$.

300 Let \mathcal{T}_1 be the class of trees with at most one branching vertex. Equivalently,
 301 \mathcal{T}_1 consists of paths and subdivisions of the claw.

302 **Proposition 1** For a fixed connected graph H , if no tree in \mathcal{T}_1 is a clique
 303 decomposition of H , then MIS in H -free graphs is $W[1]$ -hard.

304 *Proof.* This is immediate from the proof of Theorem 2 since H cannot appear
 305 in the first construction. \square

306 At this point, we can focus on connected graphs H admitting a tree $T \in \mathcal{T}_1$
 307 as a clique decomposition. The reciprocal of Proposition 1 cannot be true since
 308 a simple edge is a clique decomposition of C_4 . The next definition further
 309 restricts the interaction between two adjacent cliques.

310 **Definition 2** Let T be a graph on ℓ vertices t_1, \dots, t_ℓ . We say that T is a
 311 *strong clique decomposition of H* if there is a partition of $V(H)$ into (C_1, \dots, C_ℓ)
 312 such that:

- 313 – for each $i \in [\ell]$, $H[C_i]$ is a clique,
- 314 – for each $t_i t_j \in E(T)$, $H[C_i \cup C_j]$ is a clique, and
- 315 – for each $t_i t_j \notin E(T)$, there is no edge between C_i and C_j .

316 An equivalent way to phrase this definition is that H can be obtained
 317 from T by *adding false twins*. Adding a false twin v' to a graph consists of
 318 duplicating one of its vertex v (i.e., v and v' have the same neighbors) and
 319 then adding an edge between v and v' .

320 We define *almost strong clique decompositions* which informally are strong
 321 clique decompositions where at most one edge can be missing in the interaction
 322 between two adjacent cliques.

323 **Definition 3** Let T be a graph on ℓ vertices t_1, \dots, t_ℓ . We say that T is an
 324 *almost strong clique decomposition of H* if there is a partition of $V(H)$ into
 325 (C_1, \dots, C_ℓ) such that:

- 326 – for each $i \in [\ell]$, $H[C_i]$ is a clique,
- 327 – for each $t_i t_j \in E(T)$, $H[C_i \cup C_j]$ is a clique potentially deprived of a single
 328 edge, and is connected, and
- 329 – for each $t_i t_j \notin E(T)$, there is no edge between C_i and C_j .

330 Finally, a *nearly strong clique decomposition* is slightly weaker than an
 331 almost strong clique decomposition: at most one interaction between two ad-
 332 jacent cliques is only required to be C_4 -free. Formally:

333 **Definition 4** Let T be a graph on ℓ vertices t_1, \dots, t_ℓ with a special edge
 334 $t_a t_b$. We say that T is a *nearly strong clique decomposition of H* if there is a
 335 partition of $V(H)$ into (C_1, \dots, C_ℓ) such that:

- 336 – for each $i \in [\ell]$, $H[C_i]$ is a clique,
- 337 – $H[C_a \cup C_b]$ is C_4 -free,
- 338 – for each $t_i t_j \in E(T) \setminus \{t_a t_b\}$, $H[C_i \cup C_j]$ is a clique potentially deprived of
 339 a single edge, and is connected, and
- 340 – for each $t_i t_j \notin E(T)$, there is no edge between C_i and C_j .

341 Let \mathcal{P} be the set of all the paths. Notice that $\mathcal{T}_1 \setminus \mathcal{P}$ is the set of all the
 342 subdivisions of the claw.

343 **Theorem 3** *Let H be a fixed connected graph. If no $P \in \mathcal{P}$ is a nearly strong
 344 clique decomposition of H and no $T \in \mathcal{T}_1 \setminus \mathcal{P}$ is an almost strong clique de-
 345 composition of H , then MIS in H -free graphs is $W[1]$ -hard.*

346 *Proof.* The idea is to mainly use the second construction and the fact that MIS
 347 in C_4 -free graphs is $W[1]$ -hard (due to the first construction). For every fixed
 348 graph H which cannot be an induced subgraph in the second construction,
 349 MIS is $W[1]$ -hard. To appear in this construction, the graph H should have

- 350 – either a clique decomposition on a subdivision of the claw, such that the
 351 interaction between two adjacent cliques is the complement of a (non nec-
 352 essarily perfect) matching, or
- 353 – a clique decomposition on a path, such that the interaction between two
 354 adjacent cliques is the complement of a matching, except for at most one
 355 interaction which can be a C_4 -free graph.

356 We now just observe that in both cases if, among the interactions between
 357 adjacent cliques, one complement of matching has at least two non-edges, then
 358 H contains an induced C_4 . Hence the two items can be equivalently replaced
 359 by the existence of an almost strong clique decomposition on a subdivision of
 360 the claw, and a nearly strong clique decomposition on a path, respectively. \square

361 Theorem 3 narrows down the connected open cases to graphs H which
 362 have a nearly strong clique decomposition on a path or an almost strong clique
 363 decomposition on a subdivision of the claw.

364 In the strong clique decomposition, the interaction between two adjacent
 365 cliques is very simple: their union is a clique. Therefore, it might be tempting
 366 to conjecture that if H admits $T \in \mathcal{T}_1$ as a strong clique decomposition, then
 367 MIS in H -free graphs is FPT. Indeed, those graphs H appear in both the first
 368 and the second $W[1]$ -hardness constructions. Nevertheless, we will see that
 369 this conjecture is false: even if H has a strong clique decomposition $T \in \mathcal{T}_1$,
 370 it can be that MIS is $W[1]$ -hard. The simplest tree of $\mathcal{T}_1 \setminus \mathcal{P}$ is the claw. We
 371 denote by $T_{i,j,k}$ the graph obtained by adding a universal vertex to the disjoint
 372 union of three cliques $K_i \uplus K_j \uplus K_k$. The claw is a strong clique decomposition
 373 of $T_{i,j,k}$ (for every natural numbers i, j, k).

374 **Theorem 4** MIS in $T_{1,2,2}$ -free graphs is $W[1]$ -hard.

375 *Proof.* We show that $T_{1,2,2}$ does not appear in the third construction (Fig-
 376 ure 4, right). We claim that, in this construction, the graph $T_{1,1,2}$, sometimes
 377 called cricket, can only appear in the two ways depicted on Figure 5 (up to
 378 symmetry).

379 **Claim 5.** *The triangle of the cricket cannot appear within the same main*
 380 *clique.*

381 *Proof of claim:* Otherwise the two leaves (*i.e.*, vertices of degree 1) of the
 382 cricket are in two distinct adjacent cliques. But at least one of those adjacent
 383 cliques is linked to the main clique of the triangle by an anti-matching. This
 384 is a contradiction to the corresponding leaf having two non-neighbors in the
 385 main clique of the triangle. \triangleleft

386 We first study how the cricket can appear in a path of cliques. Let C be
 387 the main clique containing the universal vertex of the cricket. This vertex is
 388 adjacent to three disjoint cliques $K_1 \uplus K_1 \uplus K_2$. Due to the previous claim, the
 389 only way to distribute them is to put K_1 in the previous main clique, K_1 in the
 390 same main clique C , and K_2 in the next main clique. This is only possible if the
 391 interaction between C and the next main clique is a half-graph. In particular,
 392 this implies that the interaction between the previous main clique and C is an
 393 anti-matching. This situation corresponds to the left of Figure 5.

394 This also implies that the cricket cannot appear in a path of cliques without
 395 a half-graph interaction (anti-matchings only). We now turn our attention
 396 to the vicinity of a triangle of main cliques, which is proper to the third
 397 construction. By our previous remarks, we know that the universal vertex of

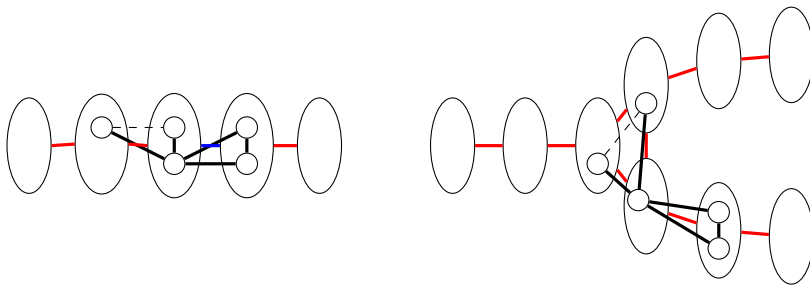


Fig. 5 The two ways the cricket appears in the third construction. The red edges between two adjacent cliques symbolize an anti-matching, whereas the blue edge symbolizes a C_4 -free graph. In the left hand-side, one neighbor of the universal vertex with degree 2 could alternatively be in the same clique as the universal vertex.

398 the cricket has to be either alone in a main clique (by symmetry, it does not
 399 matter which one) of the triangle, or with exactly one of its neighbors of degree
 400 2. Now, the only way to place $K_1 \uplus K_1 \uplus K_2$ is to put the two K_1 in the two
 401 other main cliques of the triangle, and the K_2 (or the single vertex rest of it) in
 402 the remaining adjacent main clique. Indeed, if the K_2 is in a main clique of the
 403 triangle, the K_1 in the third main clique of the triangle would have two non-
 404 edges towards to K_2 . This is not possible with an anti-matching interaction.
 405 Therefore, the only option corresponds to the right of Figure 5.

406 To obtain a $T_{1,2,2}$, one needs to find a false twin to one of the leaves of
 407 the cricket. This is not possible since, in both cases, the two leaves are in two
 408 adjacent cliques with an anti-matching interaction. Therefore, adding the false
 409 twin would create a second non-neighbor to the remaining leaf. \square

410 The graph $T_{1,1,1}$ is the claw itself for which MIS is solvable in polynomial
 411 time. The parameterized complexity for the graph $T_{1,1,2}$ (the cricket) remains
 412 open. As a matter of fact, this question is unresolved for $T_{1,1,s}$ -free graphs,
 413 for any integer $s \geq 2$. Solving those cases would bring us a bit closer to a
 414 full dichotomy *FPT vs W[1]-hard*. Although, Theorem 4 suggests that this
 415 dichotomy will be rather subtle. In addition, this result infirms the plausible
 416 conjecture: *if MIS is FPT in H-free graphs, then it is FPT in H'-free graphs*
 417 *where H' can be obtained from H by adding false twins.*

418 The toughest challenge towards the dichotomy is understanding MIS in
 419 the absence of *paths of cliques*⁵. In Theorem 11, we make a very first step
 420 in that direction: we show that for every graph H with a strong clique decom-
 421 position on P_3 , the problem is FPT. In the previous paragraphs, we dealt
 422 mostly with connected graphs H . In Theorem 6, we show that if H is a disjoint
 423 union of cliques, then MIS in H -free graphs is FPT. In the language of clique
 424 decompositions, this can be phrased as *H has a clique decomposition on an*
 425 *edgeless graph.*

⁵ Actually, even the classical complexity of MIS in the absence of long induced paths is not well understood

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 proof of the following theorem is inspired by the case $r = 2$ by Alekseev [2].

Theorem 6 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 $Ram(r, k)^{qk} n^{qr}$ 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. We also assume $r \geq 3$, since the case $r = 2$ corresponds to Alekseev's algorithm[2].

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). Since a clique of size r can be found in polynomial time, such an ordering can be found in polynomial time. For any vertex x , define $x^+ = \{y, x < y\}$ and $x^- = V(G) \setminus x^+$.

Let us first explain how we will generate independent sets. We will prove next that the algorithm generates all of them. 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 $Ram(r, k)^{(q-1)k} n^{(q-1)r}$ 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 . For every choice of a vertex x amongst the largest $Ram(r, k)$ 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 non adjacent to x . We repeat this operation $k - 1$ times (or until V_2 becomes empty). At the end, we either find an independent set of size k (if V_2 is still not empty) or add S_1 to \mathcal{S}_C (when V_2 becomes empty). By doing so we construct a family of at most $Ram(r, k)^k$ independent sets for each S_1 , so in total we get indeed at most $Ram(r, k)^{kq} n^{(q-1)r}$ 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 $S \cap c_{last}^+$

469 is empty and thus the first item is vacuously satisfied. Secondly, note that
 470 several cliques C might satisfy the two previous conditions. In that case, pick
 471 one such clique arbitrarily. This definition of C and c ensures that $S \cap c^+$ is an
 472 independent set in the set V_1 defined in the construction above (it might be
 473 empty). Thus, it will be picked in the second phase as some S_1 in \mathcal{S}_1 and for
 474 this choice, each time V_2 is considered, the fact that C is chosen to minimize its
 475 largest element c guarantees that there must be a vertex of S in the $Ram(r, k)$
 476 largest vertices in V_2 , otherwise we could find within those vertices an r -clique
 477 contradicting the choice of C (we can find an r -clique satisfying both points
 478 such that the maximum vertex is smaller than c). So it ensures that we will
 479 add S to the collection \mathcal{S}_C , which concludes our proof. \square

480 4 Positive results II

481 4.1 Key ingredient: Iterative expansion and Ramsey extraction

482 In this section, we present the main idea of our algorithms. It is a general-
 483 ization of iterative expansion, which itself is the maximization version of the
 484 well-known iterative compression technique. Iterative compression is a useful
 485 tool for designing parameterized algorithms for subset problems (*i.e.* problems
 486 where a solution is a subset of some set of elements: vertices of a graph, vari-
 487 ables of a logic formula...*etc.*) [10, 25]. Although it has been mainly used for
 488 minimization problems, iterative compression has been successfully applied
 489 for maximization problems as well, under the name *iterative expansion* [7].
 490 Roughly speaking, when the problem consists of finding a solution of size at
 491 least k , the iterative expansion technique consists of solving the problem where
 492 a solution S of size $k - 1$ is given in the input, in the hope that this solution
 493 will imply some structure in the instance. In the following, we consider an
 494 extension of this approach where, instead of a single smaller solution, one is
 495 given a set of $f(k)$ smaller solutions $S_1, \dots, S_{f(k)}$. As we will see later, we can
 496 further add more constraints on the sets $S_1, \dots, S_{f(k)}$. Notice that all the re-
 497 sults presented in this sub-section (Lemmas 2 and 3 in particular) hold for any
 498 hereditary graph class (including the class of all graphs). The use of properties
 499 inherited from particular graphs (namely, H -free graphs in our case) will only
 500 appear in Sections 4.2 and 4.3.

501 **Definition 5** For a function $f : \mathbb{N} \rightarrow \mathbb{N}$, the f -ITERATIVE EXPANSION MIS
 502 problem takes as input a graph G , an integer k , and a set of $f(k)$ vertex-
 503 disjoint independent sets $S_1, \dots, S_{f(k)}$, each of size $k - 1$. The objective is to
 504 find an independent set of size k in G , or to decide that such an independent
 505 set does not exist.

506 **Lemma 2** Let \mathcal{G} be a hereditary graph class. MIS is FPT in \mathcal{G} iff f -ITERATIVE
 507 EXPANSION MIS is FPT in \mathcal{G} for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$.

508 *Proof.* Clearly if MIS is FPT, then f -ITERATIVE EXPANSION MIS is FPT
 509 for any computable function f . Conversely, let f be a function for which f -
 510 ITERATIVE EXPANSION MIS is FPT, and let G be a graph with $|V(G)| = n$.

511 We show by induction on k that there is an algorithm that either finds an
 512 independent set of size k , or answers that such a set does not exist, in FPT time
 513 parameterized by k . The initialization can obviously be computed in constant
 514 time. Assume we have an algorithm for $k - 1$. Successively for i from 1 to $f(k)$,
 515 we construct an independent set S_i of size $k - 1$ in $G \setminus (S_1, \dots, S_{i-1})$. If, for
 516 some i , we are unable to find such an independent set, then it implies that any
 517 independent set of size k in G must intersect $S_1 \cup \dots \cup S_{i-1}$. We thus branch on
 518 every vertex v of this union, and, by induction, find an independent set of size
 519 $k - 1$ in the graph induced by $V(G) \setminus N[v]$. If no step i triggered the previous
 520 branching, we end up with $f(k)$ vertex-disjoint independent sets $S_1, \dots, S_{f(k)}$,
 521 each of size $k - 1$. We now invoke the algorithm for f -ITERATIVE EXPANSION
 522 MIS to conclude. Let us analyze the running time of this algorithm: each
 523 step either branches on at most $f(k)(k - 1)$ subcases with parameter $k - 1$, or
 524 concludes in time $\mathcal{A}_f(n, k)$, the running time of the algorithm for f -ITERATIVE
 525 EXPANSION MIS. Hence the total running time is $O^*(f(k)^k(k - 1)^k \mathcal{A}_f(n, k))$,
 526 where the $O^*(\cdot)$ suppresses polynomial factors. \square

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

531 **Definition 6** Given a graph G and a set of $k - 1$ vertex-disjoint cliques of G ,
 532 $\mathcal{C} = \{C_1, \dots, C_{k-1}\}$, each of size q , we say that \mathcal{C} is a set of *Ramsey-extracted*
 533 *cliques of size q* if the conditions below hold. Let $C_r = \{c_j^r : j \in \{1, \dots, q\}\}$
 534 for every $r \in \{1, \dots, k - 1\}$.

- 535 – For every $j \in [q]$, the set $\{c_j^r : r \in \{1, \dots, k - 1\}\}$ is an independent set of
 536 G of size $k - 1$.
- 537 – For any $r \neq r' \in \{1, \dots, k - 1\}$, one of the four following case can happen:
 538 (i) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \notin E(G)$
 539 (ii) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j \neq j'$
 540 (iii) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j < j'$
 541 (iv) for every $j, j' \in [q]$, $c_j^r c_{j'}^{r'} \in E(G)$ iff $j > j'$

542 In the case (i) (resp. (ii)), we say that the relation between C_r and $C_{r'}$ is
 543 *empty* (resp. *full*⁶). In case (iii) or (iv), we say the relation is *semi-full*.

544 Observe, in particular, that a set \mathcal{C} of $k - 1$ Ramsey-extracted cliques of
 545 size q can be partitioned into q independent sets of size $k - 1$. As we will see
 546 later, these cliques will allow us to obtain more structure with the remaining
 547 vertices if the graph is H -free. Roughly speaking, if q is large, we will be able to

⁶ Remark that in this case, the graph induced by $C_r \cup C_{r'}$ is the complement of a perfect matching.

548 extract from \mathcal{C} another set \mathcal{C}' of $k-1$ Ramsey-extracted cliques of size $q' < q$,
 549 such that every clique is a module⁷ with respect to the solution x_1^*, \dots, x_k^* we
 550 are looking for. Then, by guessing the structure of the adjacencies between \mathcal{C}'
 551 and the solution, we will be able to identify from the remaining vertices k sets
 552 X_1, \dots, X_k , where each X_i has the same neighborhood as x_i^* w.r.t. \mathcal{C}' , and
 553 plays the role of “candidates” for this vertex. For a function $f : \mathbb{N} \rightarrow \mathbb{N}$, we
 554 define the following problem:

555 **Definition 7** The f -RAMSEY-EXTRACTED ITERATIVE EXPANSION MIS prob-
 556 lem takes as input an integer k and a graph G whose vertices are partitioned
 557 into non-empty sets $X_1 \cup \dots \cup X_k \cup C_1 \cup \dots \cup C_{k-1}$, where:

- 558 – $\{C_1, \dots, C_{k-1}\}$ is a set of $k-1$ Ramsey-extracted cliques of size $f(k)$
- 559 – any independent set of size k in G is contained in $X_1 \cup \dots \cup X_k$
- 560 – $\forall i \in \{1, \dots, k\}, \forall v, w \in X_i$ and $\forall j \in \{1, \dots, k-1\}, N(v) \cap C_j = N(w) \cap$
 561 $C_j = \emptyset$ or $N(v) \cap C_j = N(w) \cap C_j = C_j$
- 562 – the following bipartite graph \mathcal{B} is connected: $V(\mathcal{B}) = B_1 \cup B_2, B_1 =$
 563 $\{b_1^1, \dots, b_k^1\}, B_2 = \{b_1^2, \dots, b_{k-1}^2\}$ and $b_j^1 b_r^2 \in E(\mathcal{B})$ iff X_j and C_r are adja-
 564 cent.

565 The objective is the following:

- 566 – if G contains an independent set S such that $S \cap X_i \neq \emptyset$ for all $i \in$
 567 $\{1, \dots, k\}$, then the algorithm must answer “YES”. In that case the solu-
 568 tion is called a *rainbow independent set*.
- 569 – if G does not contain an independent set of size k , then the algorithm must
 570 answer “NO”.

571 Observe that in the case the graph contains an independent set of size k but
 572 no rainbow independent set, the algorithm is allowed to answer either *yes* or
 573 *no*. Eventually, this will imply a one-sided error Monte-Carlo algorithm with
 574 constant error probability for MIS. Definition 7 is illustrated by Figure 6.

575 **Lemma 3** Let \mathcal{G} be a hereditary graph class. If there exists a computable func-
 576 tion $f : \mathbb{N} \rightarrow \mathbb{N}$ such that f -RAMSEY-EXTRACTED ITERATIVE EXPANSION
 577 MIS is FPT in \mathcal{G} , then g -ITERATIVE EXPANSION MIS is FPT in \mathcal{G} , where
 578 $g(x) = \text{Ram}_{\ell_x}(f(x)2^{x(x-1)}) \forall x \in \mathbb{N}$, with $\ell_x = 2^{(x-1)^2}$.

579 *Proof.* Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be such a function, and let G, k and $\mathcal{S} = \{S_1, \dots, S_{g(k)}\}$
 580 be an input of g -ITERATIVE EXPANSION MIS. Recall that the objective is to
 581 find an independent set of size k in G , or to decide that $\alpha(G) < k$. We prove it
 582 by induction on k . If G contains an independent set of size k , then either there
 583 is one intersecting some set of \mathcal{S} , or every independent set of size k avoids the
 584 sets in \mathcal{S} . In order to capture the first case, we branch on every vertex v of
 585 the sets in \mathcal{S} , and make a recursive call with parameter $G \setminus N[v], k-1$. In the
 586 remainder of the algorithm, we thus assume that any independent set of size
 587 k in G avoids every set of \mathcal{S} .

⁷ A set of vertices M is a module if every vertex $v \notin M$ is adjacent to either all vertices of M , or none.

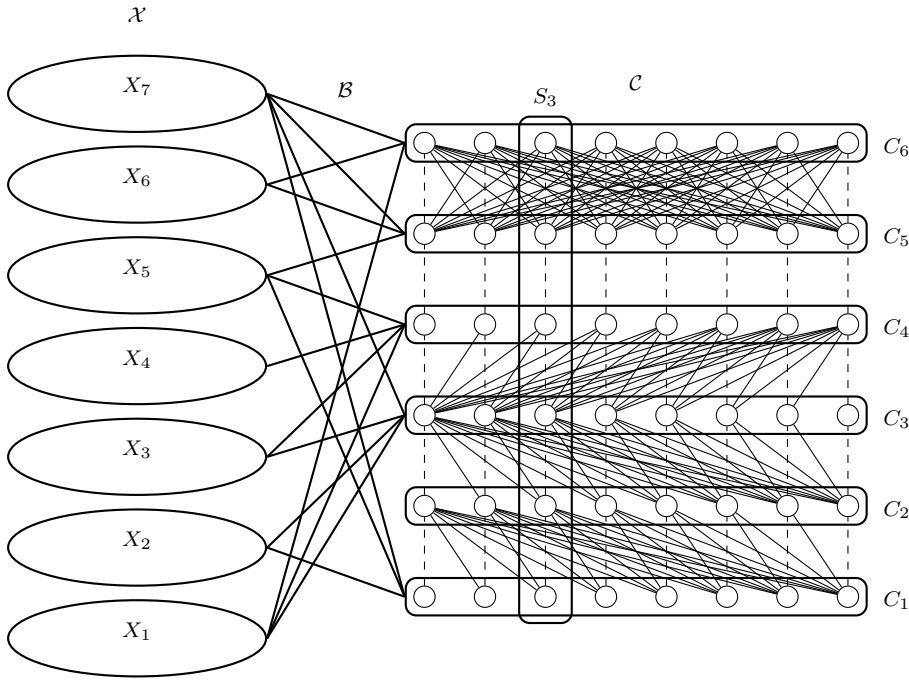


Fig. 6 The structure of the f -RAMSEY-EXTRACTED ITERATIVE EXPANSION MIS inputs.

588 We choose an arbitrary ordering of the vertices of each S_j . Let us denote
589 by s_j^r the r^{th} vertex of S_j . Notice that given an ordered pair of sets of $k-1$
590 vertices (A, B) , there are $\ell_k = 2^{\binom{k-1}{2}}$ possible sets of edges between these two
591 sets. Let us denote by $c_1, \dots, c_{2^{\binom{k-1}{2}}}$ the possible sets of edges, called *types*.
592 We define an auxiliary edge-colored graph H whose vertices are in one-to-one
593 correspondence with $S_1, \dots, S_{g(k)}$, and, for $i < j$, there is an edge between S_i
594 and S_j of color γ iff the type of (S_i, S_j) is γ . By Ramsey's theorem, since H has
595 $\text{Ram}_{\ell_k}(f(k)2^{k(k-1)})$ vertices, it must admit a monochromatic clique of size at
596 least $h(k) = f(k)2^{k(k-1)}$. *W.l.o.g.*, the vertex set of this clique corresponds to
597 $S_1, \dots, S_{h(k)}$. For $p \in \{1, \dots, k-1\}$, let $C_p = \{s_1^p, \dots, s_{h(k)}^p\}$. Observe that the
598 Ramsey extraction ensures that each C_p is either a clique or an independent
599 set. If C_p is an independent set for some p , then we can immediately conclude,
600 since $h(k) \geq k$. Hence, we suppose that C_p is a clique for every $p \in \{1, \dots, k-1\}$.
601 We now prove that C_1, \dots, C_{k-1} are Ramsey-extracted cliques of size $h(k)$.
602 First, by construction, for every $j \in \{1, \dots, h(k)\}$, the set $\{s_j^p : p = 1, \dots, k-1\}$
603 is an independent set. Then, let c be the type of the monochromatic clique
604 of H obtained previously, represented by the adjacencies between two sets
605 (A, B) , each of size $k-1$. For every $p \in \{1, \dots, k-1\}$, let a_p (resp. b_p) be the
606 p^{th} vertex of A (resp. B). Let $p, q \in \{1, \dots, k-1\}$, $p \neq q$. If none of $a_p b_q$ and
607 $a_q b_p$ are edges in type c , then there is no edge between C_p and C_q , and their
608 relation is thus empty. If both edges $a_p b_q$ and $a_q b_p$ exist in c , then the relation

609 between C_p and C_q is full. Finally if exactly one edge among $a_p b_q$ and $a_q b_p$
 610 exists in c , then the relation between C_p and C_q is semi-full. This concludes
 611 the fact that $\mathcal{C} = \{C_1, \dots, C_{k-1}\}$ are Ramsey-extracted cliques of size $h(k)$.

612 Suppose that G has an independent set $X^* = \{x_1^*, \dots, x_k^*\}$. Recall that
 613 we assumed previously that X^* is contained in $V(G) \setminus (C_1 \cup \dots \cup C_{k-1})$. The
 614 next step of the algorithm consists of branching on every subset of $f(k)$ indices
 615 $J \subseteq \{1, \dots, h(k)\}$, and restrict every set C_p to $\{s_j^p : j \in J\}$. For the sake
 616 of readability, we keep the notation C_p to denote $\{s_j^p : j \in J\}$ (the non-
 617 selected vertices are put back in the set of remaining vertices of the graph,
 618 *i.e.* we do not delete them). Since $h(k) = f(k)2^{k(k-1)}$, there must exist a
 619 branch where the chosen indices are such that for every $i \in \{1, \dots, k\}$ and
 620 every $p \in \{1, \dots, k-1\}$, x_i^* is either adjacent to all vertices of C_p or none
 621 of them. In the remainder, we may thus assume that such a branch has been
 622 made, with respect to the considered solution $X^* = \{x_1^*, \dots, x_k^*\}$. Now, for
 623 every $v \in V(G) \setminus (C_1, \dots, C_{k-1})$, if there exists $p \in \{1, \dots, k-1\}$ such that
 624 $N(v) \cap C_p \neq \emptyset$ and $N(v) \cap C_p \neq C_p$, then we can remove this vertex, as
 625 we know that it cannot correspond to any x_i^* . Thus, we know that all the
 626 remaining vertices v are such that for every $p \in \{1, \dots, k-1\}$, v is either
 627 adjacent to all vertices of C_p , or none of them.

628 In the following, we perform a color coding-based step on the remaining ver-
 629 tices. Informally, this color coding will allow us to identify, for every vertex x_i^*
 630 of the optimal solution, a set X_i of candidates, with the property that all ver-
 631 tices in X_i have the same neighborhood with respect to sets C_1, \dots, C_{k-1} . We
 632 thus color uniformly at random the remaining vertices $V(G) \setminus (C_1, \dots, C_{k-1})$
 633 using k colors. The probability that the elements of X^* are colored with pair-
 634 wise distinct colors is at least e^{-k} .

635 This random process can be derandomized using the so-called notion of
 636 perfect hash families. A (n, k) -perfect hash family is a family of functions
 637 \mathcal{F} from $[n]$ to $[k]$ (which can be seen as colorings) such that for every set
 638 $S \in \binom{[n]}{k}$, there exists $f \in \mathcal{F}$ such that the restriction of f on S is injective.
 639 It is known [23] that a (n, k) -perfect hash family of size $e^k k^{O(\log k)} \log n$ can
 640 be constructed in time $e^k k^{O(\log k)} n \log n$. Hence, instead of coloring $V(G) \setminus$
 641 (C_1, \dots, C_{k-1}) uniformly at random, we branch on every coloring $f \in \mathcal{F}$ and
 642 run the remainder of the algorithm. The definition of (n, k) -perfect hash family
 643 ensures that there is a coloring f such that X^* is a rainbow independent set
 644 with respect to f . Notice that this derandomization step implies a branching
 645 into $h(k) \log n$ subcases, for some computable function h . However, the depth
 646 of the branching tree (*i.e.* the maximum number of times this branching will
 647 be made in every computation path) is bounded by a function of k only. Since
 648 $(\log n)^k \leq g(k)n$ for some function g [26], the deterministic version of the
 649 algorithm is still FPT.

650 We are thus reduced to the case of finding a rainbow independent set. For
 651 every $i \in \{1, \dots, k\}$, let X_i be the vertices of $V(G) \setminus (C_1, \dots, C_{k-1})$ colored
 652 with color i . We now partition every set X_i into at most 2^{k-1} subsets $X_i^1, \dots,$
 653 $X_i^{2^{k-1}}$, such that for every $j \in \{1, \dots, 2^{k-1}\}$, all vertices of X_i^j have the same

neighborhood with respect to the sets C_1, \dots, C_{k-1} (recall that every vertex of $V(G) \setminus (C_1, \dots, C_{k-1})$ is adjacent to all vertices of C_p or none, for each $p \in \{1, \dots, k-1\}$). We branch on every tuple $(j_1, \dots, j_k) \in \{1, \dots, 2^{k-1}\}$. Clearly the number of branches is bounded by a function of k only and, moreover, one branch (j_1, \dots, j_k) is such that x_i^* has the same neighborhood in $C_1 \cup \dots \cup C_{k-1}$ as vertices of $X_i^{j_i}$ for every $i \in \{1, \dots, k\}$. We assume in the following that such a branching has been made. For every $i \in \{1, \dots, k\}$, 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) , $B_1 = \{b_1^1, \dots, b_k^1\}$, $B_2 = \{b_1^2, \dots, b_{k-1}^2\}$, 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, \dots, C_{k-1} , this bipartite graph actually corresponds to the one described in Definition 7 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 $C_1 \cup \dots \cup C_{k-1}$ 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 7 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 $i \in \{1, \dots, k\}$. We thus run an algorithm solving f -RAMSEY-EXTRACTED ITERATIVE EXPANSION MIS on this input, which concludes the algorithm. \square

The proof of the following result is immediate, by using successively Lemmas 2 and 3.

Theorem 7 *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 8 *For any $r \geq 2$ and $2 \leq s < r$, MIS in $(K_r \setminus K_s)$ -free graphs is FPT if $s \leq 3$, and $W[1]$ -hard otherwise.*

Proof. The case $s = 2$ was already known [12]. The result for $s \geq 4$ comes from Theorem 2. We now deal with the case $s = 3$. We solve the problem in $(K_{r+3} \setminus K_3)$ -free graphs, for every $r \geq 2$ (the problem is polynomial for $r = 1$, since it corresponds exactly to the case of claw-free graphs). Let G, k be an input of the problem. We present an FPT algorithm for f -RAMSEY-EXTRACTED ITERATIVE EXPANSION MIS with $f(x) = r$ for every $x \in \mathbb{N}$. The result for MIS can then be obtained using Theorem 7.

We thus assume that $V(G) = X_1 \cup \dots \cup X_k \cup C_1 \cup \dots \cup C_{k-1}$ where all cliques C_p have size r . Consider the bipartite graph \mathcal{B} representing the

adjacencies between $\{X_1, \dots, X_k\}$ and $\{C_1, \dots, C_{k-1}\}$, as in Definition 7 (for the sake of readability, we will make no distinction between the vertices of \mathcal{B} and the sets $\{X_1, \dots, X_k\}$ and $\{C_1, \dots, C_{k-1}\}$). We may first assume that $|X_i| \geq \text{Ram}(r, k)$ for every $i \in \{1, \dots, k\}$, since otherwise we can branch on every vertex v of X_i and make a recursive call with input $G \setminus N[v]$, $k - 1$. Hence, for every $i \in \{1, \dots, k\}$, we may assume that X_i contains a clique on r vertices (indeed, if it does not, then it must contain an independent set of size k , in which case we are done). Suppose now that G contains an independent set $S^* = \{x_1^*, \dots, x_k^*\}$, with $x_i \in X_i$ for all $i \in \{1, \dots, k\}$. The first step is to consider the structure of \mathcal{B} , using the fact that G is $(K_r \setminus K_3)$ -free. We have the following:

Claim 9. \mathcal{B} is a path, or we can conclude in polynomial time.

Proof of claim: We first prove that for every $i \in \{1, \dots, k\}$, the degree of X_i in \mathcal{B} is at most 2. Indeed, assume by contradiction that it is adjacent to C_a , C_b and C_c . Since $|X_i| \geq \text{Ram}(r, k)$, by Ramsey's theorem, it either contains an independent set of size k , in which case we are done, or a clique K of size r . However, observe in this case that K together with s_1^a , s_1^b and s_1^c (which are pairwise non-adjacent) induces a graph isomorphic to $K_{r+3} \setminus K_3$.

Then, we show that for every $i \in \{1, \dots, k - 1\}$, the degree of C_i in \mathcal{B} is at most 2. Assume by contradiction that C_i is adjacent to X_a , X_b and X_c . If the instance is positive, then there must be an independent set of size three with non-empty intersection with each of X_a , X_b and X_c . If such an independent set does not exist (which can be checked in cubic time), we can immediately answer NO. Now observe that C_i (which is of size r) together with this independent set induces a graph isomorphic to $K_{r+3} \setminus K_3$.

To summarize, \mathcal{B} is a connected bipartite graph of maximum degree 2 with k vertices in one part, $k - 1$ vertices in the other part. It must be a path. \triangleleft

W.l.o.g., we may assume that for every $i \in \{2, \dots, k - 1\}$, X_i is adjacent to C_{i-1} and C_i , and that X_1 (resp. X_k) is adjacent to C_1 (resp. C_{k-1}). We now concentrate on the adjacencies between sets X_i . We say that an edge $xy \in E(G)$ is a *long edge* if $x \in X_i$, $y \in X_j$ with $|j - i| \geq 2$ and $2 \leq i, j \leq k - 1$, $i \neq j$.

Claim 10. $\forall x \in X_2 \cup \dots \cup X_{k-1}$, x is incident to at most $(k - 2)(\text{Ram}(r, 3) - 1)$ long edges.

Proof of claim: In order to prove it, let us show that for $i, j \in \{2, \dots, k - 1\}$ such that $|j - i| \geq 2$, $i \neq j$, and for every $x \in X_i$, $|N(x) \cap X_j| \leq \text{Ram}(r, 3) - 1$. Assume by contradiction that there exists $x \in X_i$ which has at least $\text{Ram}(r, 3)$ neighbors $Y \subseteq X_j$. By Ramsey's theorem, either Y contains an independent set of size 3 or a clique of size r . In the first case, C_j together with these three vertices induces a graph isomorphic to $K_{r+3} \setminus K_3$. Hence we may assume that Y contains a clique Y' of size r . But in this case, Y' together with x , s_1^{j-1} , s_1^j induce a graph isomorphic to $K_{r+3} \setminus K_3$ as well. \triangleleft

737 Recall that the objective is to find an independent set of size k with non-
 738 empty intersection with X_i , for every $i \in \{1, \dots, k\}$. We assume $k \geq 5$, other-
 739 wise the problem is polynomial. The algorithm starts by branching on every
 740 pair of non-adjacent vertices $(x_1, x_k) \in X_1 \times X_k$, and removing the union of
 741 their neighborhoods in $X_2 \cup \dots \cup X_{k-1}$. For the sake of readability, we still
 742 denote by X_2, \dots, X_{k-1} these reduced sets. If such a pair does not exist or
 743 the removal of their neighborhood empties some X_i , then we immediately answer
 744 NO (for this branch). Informally speaking, we just guessed the solution
 745 within X_1 and X_k (the reason for this is that we cannot bound the number
 746 of long edges incident to vertices of these sets). We now concentrate on the
 747 graph G' , which is the graph induced by $X_2 \cup \dots \cup X_{k-1}$. Clearly, it remains
 748 to decide whether G' admits an independent set of size $k - 2$ with non-empty
 749 intersection with X_i , for every $i \in \{2, \dots, k - 1\}$.

750 The previous claim showed that the structure of G' is quite particular:
 751 roughly speaking, the adjacencies between consecutive X_i 's is arbitrary, but
 752 the number of long edges is bounded for every vertex. The key observation is
 753 that if there were no long edge at all, then a simple dynamic programming
 754 algorithm would allow us to conclude. Nevertheless, using the previous claim,
 755 we can actually upper bound the number of long edges incident to a vertex
 756 of the solution by a function of k only (recall that r is a constant). We can
 757 then get rid of these problematic long edges using the so-called technique of
 758 *random separation* [6]. Let $S = \{x_2, \dots, x_{k-1}\}$ be a solution of our problem
 759 (with $x_i \in X_i$ for every $i \in \{2, \dots, k - 1\}$). Let us define $D = \{y : xy \text{ is a long}$
 760 $\text{edge and } x \in S\}$. By the previous claim, we have $|D| \leq (Ram(r, 3) - 1)(k - 2)^2$.
 761 The idea of random separation is to delete each vertex of the graph with
 762 probability $\frac{1}{2}$. At the end, we say that a removal is *successful* if both of the
 763 two following conditions hold: (i) no vertex of S has been removed, and (ii)
 764 all vertices of D have been removed (other vertices but S may have also been
 765 removed). Observe that the probability that a removal is successful is at least
 766 $2^{-k^2 Ram(r, 3)}$. In such a case, we can remove all remaining long edges (more
 767 formally, we remove their endpoints): indeed, for a remaining long edge xy , we
 768 know that there exists a solution avoiding both x and y , hence we can safely
 769 delete x and y .

770 Similarly to the color coding step of Lemma 3, this can be derandomized
 771 using (n, t) -universal sets: a (n, t) -*universal set* is a family \mathcal{U} of subsets of $[n]$
 772 such that for any $S \subseteq [n]$ of size t , the family $\{A \cap S : A \in \mathcal{U}\}$ contains all
 773 2^t subsets of S . It is known [23] that for any $n, t \geq 1$, one can construct an
 774 (n, t) -universal set of size $2^t t^{O(\log t)} \log n$ in time $2^t t^{O(\log t)} n \log n$. Let \mathcal{U} be
 775 an (n, t) -universal set for $t = k + (Ram(r, 3) - 1)(k - 2)^2$. Instead of deleting
 776 vertices of G randomly, branch on every set $U \in \mathcal{U}$, and for each branch, delete
 777 vertices from U . Then there must be a branch where $D \subseteq U$ and $S \not\subseteq U$, hence
 778 vertices of D are deleted while those of S are not. As previously, this implies
 779 branching into $h(k) \log n$ subcases for some computable function of k , but since
 780 the depth of the branching tree is a function of k only, the running time of the
 781 deterministic version is still FPT.

782 We still denote by X_2, \dots, X_{k-1} the reduced sets, for the sake of read-
 783 ability. We thus end up with a graph composed of sets X_2, \dots, X_{k-1} , with
 784 edges between X_i and X_j only if $|j - i| = 1$. In that case, observe that there is
 785 a solution if and only if the following dynamic programming returns *true* on
 786 input $P(3, x_2)$ for some $x_2 \in X_2$:

$$P(i, x_{i-1}) = \begin{cases} \text{true} & \text{if } i = k \\ \text{false} & \text{if } X_i \subseteq N(x_{i-1}) \\ \bigvee_{x_i \in X_i \setminus N(x_{i-1})} P(i+1, x_i) & \text{otherwise.} \end{cases}$$

787 Informally, this dynamic programming relies on the fact that the only ad-
 788 jacencies between sets X_i are between consecutive sets, hence we only need
 789 to remember the previous choice when constructing a solution from $i = 2$ to
 790 $k - 1$. Hence, $P(i, x_{i-1})$ represents whether there exists a rainbow solution in
 791 $\bigcup_{j=i-1}^{k-1} X_j$ containing $x_{i-1} \in X_{i-1}$. Clearly this dynamic programming runs
 792 in $O(mnk)$ time, where m and n are the number of edges and vertices of
 793 the remaining graph, respectively. Moreover, it can easily be turned into an
 794 algorithm returning a solution of size $k - 2$ if it exists. □

795

796 4.3 Clique minus a complete bipartite graph

797 For every three positive integers r, s_1, s_2 with $s_1 + s_2 < r$, we consider the
 798 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
 799 $s_1, r - s_1 - s_2$, and s_2 . More formally, every graph $K_r \setminus K_{s_1, s_2}$ can be obtained
 800 from a P_3 by adding $s_1 - 1$ false twins of the first vertex, $r - s_1 - s_2 - 1$, for
 801 the second, and $s_2 - 1$, for the third.

802 **Theorem 11** *For any $r \geq 2$ and $s_1 \leq s_2$ with $s_1 + s_2 < r$, MIS in $K_r \setminus K_{s_1, s_2}$ -*
 803 *free graphs is FPT.*

804 *Proof.* It is more convenient to prove the result for $K_{3r} \setminus K_{r, r}$ -free graphs, for
 805 any positive integer r . It implies the theorem by choosing this new r to be
 806 larger than s_1, s_2 , and $r - s_1 - s_2$. We will show that for $f(x) := 3r$ for every
 807 $x \in \mathbb{N}$, f -RAMSEY-EXTRACTED ITERATIVE EXPANSION MIS in $K_{3r} \setminus K_{r, r}$ -free
 808 graphs is FPT. By Theorem 7, this implies that MIS is FPT in this class. Let
 809 C_1, \dots, C_{k-1} (whose union is denoted by \mathcal{C}) be the Ramsey-extracted cliques
 810 of size $3r$, which can be partitioned, as in Definition 7, into $3r$ independent
 811 sets S_1, \dots, S_{3r} , each of size $k - 1$. Let $\mathcal{X} = \bigcup_{i=1}^k X_i$ be the set in which we are
 812 looking for an independent set of size k . We recall that between any X_i and any
 813 C_j there are either all the edges or none. Hence, the whole interaction between
 814 \mathcal{X} and \mathcal{C} can be described by the bipartite graph \mathcal{B} described in Definition 7.
 815 Firstly, we can assume that each X_i is of size at least $Ram(r, k)$, otherwise we
 816 can branch on $Ram(r, k)$ choices to find one vertex in an optimum solution
 817 (and decrease k by one). By Ramsey's theorem, we can assume that each X_i
 818 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 6 (which is even stronger than simply solving the colored version of the problem: Theorem 6 returns YES if and only if the instance contains an independent set of size k).

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 $C_i^{s \rightarrow t} := C_i \cap \bigcup_{s \leq j \leq t} S_j$. In other words, $C_i^{s \rightarrow t}$ contains the elements of C_i having indices between s and t . Since $|C_i| = 3r$, each C_i can be partitioned into three sets, of r elements each: $C_i^{1 \rightarrow r}, C_i^{r+1 \rightarrow 2r}$ and $C_i^{2r+1 \rightarrow 3r}$. 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 $K_{3r} \setminus K_{r,r}$:

$$\begin{aligned} & - C_{i_1}^{1 \rightarrow r} \cup C_{i_2}^{r+1 \rightarrow 2r} \cup C_{i_3}^{1 \rightarrow r} \\ & - C_{i_1}^{1 \rightarrow r} \cup C_{i_2}^{r+1 \rightarrow 2r} \cup C_{i_3}^{2r+1 \rightarrow 3r} \\ & - C_{i_1}^{2r+1 \rightarrow 3r} \cup C_{i_2}^{r+1 \rightarrow 2r} \cup C_{i_3}^{1 \rightarrow r} \\ & - C_{i_1}^{2r+1 \rightarrow 3r} \cup C_{i_2}^{r+1 \rightarrow 2r} \cup C_{i_3}^{2r+1 \rightarrow 3r} \end{aligned}$$

Hence, Y is a disjoint union of cliques (since it is P_3 -free). 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 $C_i^{1 \rightarrow r} \cup C_j^{2r+1 \rightarrow 3r}$ or with $C_i^{2r+1 \rightarrow 3r} \cup C_j^{1 \rightarrow r}$, 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}

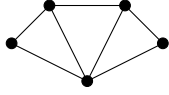


Fig. 7 The gem.

864 is $(K_r \uplus K_r)$ -free. Hence, we can run the FPT algorithm of Theorem 6 on this
 865 graph. \square

866 4.4 The gem

867 Let the *gem* be the graph obtained by adding a universal vertex to a path on
 868 four vertices (see Figure 7). Using our framework once again, we are able to
 869 obtain the following result:

870

871 **Theorem 12** *There is an FPT algorithm for MIS in gem-free graphs.*

872 *Proof.* Let $f(x) := 1$ for every $x \in \mathbb{N}$. We prove that f -RAMSEY-EXTRACTED
 873 ITERATIVE EXPANSION MIS admits an FPT algorithm in *gem*-free graphs.
 874 By the definition of f , we have $C_p = \{c_p\}$ for every $p \in \{1, \dots, k-1\}$. Recall
 875 that the objective is to find a rainbow independent set in G , or to decide that
 876 $\alpha(G) < k$. Since the bipartite graph \mathcal{B} representing the adjacencies between
 877 $\{X_1, \dots, X_k\}$ and $\{c_1, \dots, c_{k-1}\}$ is connected, it implies that for every $i \in$
 878 $\{1, \dots, k\}$, there exists $p \in \{1, \dots, k-1\}$ such that c_p dominates all vertices
 879 of X_i . Since G is *gem*-free, it implies that $G[X_i]$ is P_4 -free for every $i \in$
 880 $\{1, \dots, k\}$. Since P_4 -free graphs (*a.k.a* cographs) are perfect, the size of a
 881 maximum independent set equals the size of a clique cover. If $G[X_i]$ contains
 882 an independent set of size k (which can be tested in polynomial time), then we
 883 are done. Otherwise, we can, still in polynomial time, partition the vertices of
 884 X_i into at most $k-1$ sets $X_i^1, \dots, X_i^{q_i}$, where $G[X_i^j]$ induces a clique for every
 885 $j \in \{1, \dots, q_i\}$. We now perform a branching for every tuple (j_1, \dots, j_k) , where
 886 $j_i \in \{1, \dots, q_i\}$ for every $i \in \{1, \dots, k\}$, which, informally, allows us to guess
 887 the clique $X_i^{j_i}$ which contains the element of the rainbow independent set we
 888 are looking for. For the sake of readability, we allow ourselves this slight abuse
 889 of notation: we rename $X_i^{j_i}$ into simply X_i . Thus, for every $i \in \{1, \dots, k\}$,
 890 $G[X_i]$ is a clique.

891 Now, let $i, j \in \{1, \dots, k\}$, $i \neq j$. Let us analyse the adjacencies between X_i
 892 and X_j . We say that $\{a, b, c, d\} \subseteq X_i \cup X_j$ is a *balanced diamond* if $a, b \in X_i$
 893 ($a \neq b$), $c, d \in X_j$ ($c \neq d$) and all vertices $\{a, b, c, d\}$ are pairwise adjacent but
 894 $\{b, d\}$. We have the following claim:

895 **Claim 13.** *If the graph induced by $X_i \cup X_j$ has a balanced diamond, then X_i
 896 and X_j are twins in \mathcal{B} .*

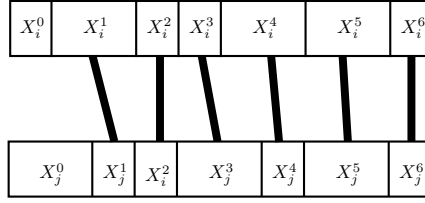


Fig. 8 Schema of the adjacencies between X_i and X_j when they do not contain a balanced diamond ($q = 6$). An edge represent a complete relation between the corresponding subsets.

897 *Proof of claim:* Suppose they are not. *W.l.o.g.* we assume that X_i is adjacent
 898 to $\{c_p\}$ while X_j is not, for some $p \in \{1, \dots, k-1\}$. Then the vertices of the
 899 balanced diamond together with c_p induce a *gem*. \triangleleft

900 The remainder of the proof consists of “cleaning” the adjacencies (X_i, X_j)
 901 having no balanced diamond (but at least one edge between them). In that
 902 case, observe that X_i and X_j can respectively be partitioned into $X_i^0, X_i^1,$
 903 \dots, X_i^q and $X_j^0, X_j^1, \dots, X_j^q$ (where X_i^0 and X_j^0 are potentially empty) such
 904 that $X_i^r \cup X_j^r$ induces a clique for every $r \in \{1, \dots, q\}$, and there is no edge
 905 between X_i^r and $X_j^{r'}$ whenever $r \neq r'$ or $r = 0$ or $r' = 0$ (see Figure 8). In
 906 each branch of the next branching rule, the sets $\{X_1, \dots, X_k\}$ will be modified
 907 into $\{X'_1, \dots, X'_k\}$. For the sake of readability, we chose to state the rule as a
 908 random one, and then explain how to derandomize it.

909

910 **Branching rule:** Let $i, j \in \{1, \dots, k\}$, $i \neq j$ such that $X_i \cup X_j$ has no
 911 balanced diamond. Then perform the following branching:

- 912 – Branch 1: $X'_i = X_i^0$ and $X'_z = X_z$ for $z \in [k] \setminus \{i\}$
- 913 – Branch 2: $X'_j = X_j^0$ and $X'_z = X_z$ for $z \in [k] \setminus \{j\}$
- 914 – Branch 3: pick a set $T \subseteq \{1, \dots, q\}$ uniformly at random, then:
 - 915 – $X'_i = \bigcup_{r \in T} X_i^r$
 - 916 – $X'_j = \bigcup_{r \notin T} X_j^r$
 - 917 – $X'_z = X_z$ for $z \in [k] \setminus \{i, j\}$

918 Consider the graph $\mathcal{G}(X_1, \dots, X_k)$ having one vertex per set X_i , and an
 919 edge between X_i and X_j if these two sets are adjacent. We now prove the
 920 following:

921 **Claim 14.** The graph $\mathcal{G}(X'_1, \dots, X'_k)$ has one edge less than $\mathcal{G}(X_1, \dots, X_k)$

922 *Proof of claim:* In all three branches, observe that there is no edge between
 923 X'_i and X'_j . \triangleleft

924 **Claim 15.** If G has no independent set of size k , then no graph obtained after
 925 the branching contains an independent set of size k .

926 *Proof of claim:* Observe that in all branches, $\bigcup_{z=1}^k X'_z \subseteq \bigcup_{z=1}^k X_z$, that is,
 927 each graph obtained in each branch is an induced subgraph of G . \triangleleft

928 **Claim 16.** *If G has a rainbow independent set, then with probability at least*
 929 *$\frac{1}{2}$, at least one branch leads to a graph having a rainbow independent set.*

930 *Proof of claim:* Suppose that G contains a rainbow independent set S^* . If
 931 S^* intersects X_i^0 , then S^* also exists in the graph of the first branch. If S^*
 932 intersects X_j^0 , then S^* also exists in the graph of the second branch. The last
 933 case is where S^* intersects $X_i^{r_1}$ and $X_j^{r_2}$, for some $r_1, r_2 \in \{1, \dots, q\}$. In that
 934 case, there is a probability of $\frac{1}{2}$ that $r_1 \in T$ and $r_2 \notin T$, which concludes the
 935 proof of the claim \triangleleft

936 The derandomization of this branching rule uses uses once again (n, t) -
 937 universal sets. However, this case is simpler since we actually need a $(q, 2)$ -
 938 universal set, which can be easily constructed as follows. For every $i \in \{1, \dots, \lceil \log q \rceil\}$,
 939 define T_i to be the set of all integers $r \leq q$ whose binary representation contains
 940 a one at the i^{th} bit. Then let $\mathcal{U} = \{T_i, i = 1 \dots \lceil \log q \rceil\}$. This family is
 941 of size $\lceil \log n \rceil$ and can be constructed in $O(n \log n)$ time. The deterministic
 942 version of the previous branching rule contains the same first two branches,
 943 and replaces the random third one by $|\mathcal{U}|$ branches, where, instead of picking
 944 $T \subseteq \{1, \dots, q\}$ at random, we branch on every $T \in \mathcal{U}$. Now, Claims 14 and
 945 15 remain the same, while Claim 16 can be replaced by the fact that if G has
 946 a rainbow independent set, then at least one branch leads to a graph having
 947 a rainbow independent set. Its correctness follows from the fact that by construction
 948 of \mathcal{U} , for every $r_1, r_2 \in \{1, \dots, q\}$, $r_1 \neq r_2$, there exists $T \in \mathcal{U}$ such
 949 that $r_1 \in T$ and $r_2 \notin T$. As in Lemma 3, this implies branching into $O(\log n)$
 950 subcases, but since the depth of the branching tree is a function of k only, the
 951 running time of the deterministic version is still FPT.

952 We apply the previous branching rule exhaustively, hence we now assume
 953 it cannot apply. For the sake of readability, we keep the notation X_1, \dots, X_k
 954 in order to denote our instance, even after an eventual application of the previous
 955 branching rule. For every X_i, X_j with $i \neq j$, there is either (i) no edge between
 956 X_i and X_j , or (ii) a balanced diamond induced by $X_i \cup X_j$. Hence, Claim 13
 957 implies that each connected component of the graph induced by $\bigcup_{i=1}^k X_i$ is a
 958 module with respect to the clique $\{c_1, \dots, c_{k-1}\}$. In particular, each connected
 959 component is dominated by some c_p , with $p \in \{1, \dots, k-1\}$, and is thus P_4 -
 960 free (otherwise, a P_4 together with this vertex c_p induce a gem), which means
 961 that we can decide in polynomial time whether G contains an independent set
 962 of size k , by deciding the problem in every connected component separately
 963 (since MIS is polynomial-time solvable in P_4 -free graphs). This concludes the
 964 proof, since by Claim 14, the previous branching rule can be applied at most
 965 $\binom{k}{2}$ times. \square

966 5 Polynomial (Turing) kernels

967 In this section we investigate some special cases of Section 4.3, in particular
 968 when H is a clique of size r minus a claw with s branches, for $s < r$. Although
 969 Theorem 11 proves that MIS is FPT for every possible values of r and s , we

970 show that when $s \geq r - 2$, the problem admits a polynomial Turing kernel,
 971 while for $s \leq 2$, it admits a polynomial kernel. Notice that the latter result
 972 is somehow tight, as Corollary 4 shows that MIS cannot admit a polynomial
 973 kernel in $(K_r \setminus K_{1,s})$ -free graphs whenever $s \geq 3$.

974 5.1 Positive results

975 The main ingredient of the two following results is a constructive version of
 976 the Erdős-Hajnal theorem for the concerned graph classes:

977 **Lemma 4 (Constructive Erdős-Hajnal for $K_r \setminus K_{1,s}$)** *For every $r \geq$
 978 2 and $s < r$, there exists a polynomial-time algorithm which takes as input
 979 a connected $(K_r \setminus K_{1,s})$ -free graph G , and constructs either a clique or an
 980 independent set of size $n^{\frac{1}{r-1}}$, where n is the number of vertices of G .*

981 *Proof.* First consider the case $s = r - 1$, i.e. the forbidden graph is K_{r-1} plus
 982 an isolated vertex. If G contains a vertex v with non-neighborhood N of size
 983 at least $n^{\frac{r-2}{r-1}}$, then, since $G[N]$ is K_{r-1} -free, by Ramsey's theorem, it must
 984 contain an independent set of size $|N|^{\frac{1}{r-2}} = n^{\frac{1}{r-1}}$, which can be found in
 985 polynomial time. We may now assume that the maximum non-degree⁸ of G
 986 is $n^{\frac{r-2}{r-1}} - 1$. We construct a clique v_1, \dots, v_q in G by picking an arbitrary
 987 vertex v_1 , removing its non-neighborhood, then picking another vertex v_2 ,
 988 removing its non-neighborhood, and repeating this process until the graph
 989 becomes empty. Using the above argument on the maximum non-degree, this
 990 process can be applied $\frac{n^{\frac{r-2}{r-1}}}{n^{\frac{1}{r-1}}} = n^{\frac{1}{r-1}}$ times, corresponding to the size of the
 991 constructed clique.

992 Now, we make an induction on $r - 1 - s$ (the base case is above). If G
 993 contains a vertex v with neighborhood N of size at least $n^{\frac{r-2}{r-1}}$, then, since $G[N]$
 994 is $(K_{r-1} \setminus K_s)$ -free, by induction it admits either a clique or an independent
 995 set of size $|N|^{\frac{1}{r-2}} = n^{\frac{1}{r-1}}$, which can be found in polynomial time. We may
 996 now assume that the maximum degree of G is $n^{\frac{r-2}{r-1}} - 1$. We construct an
 997 independent set v_1, \dots, v_q in G by picking an arbitrary vertex v_1 , removing
 998 its neighborhood, and repeating this process until the graph becomes empty.
 999 Using the above argument on the maximum degree, this process can be applied
 1000 $\frac{n^{\frac{r-2}{r-1}}}{n^{\frac{1}{r-1}}} = n^{\frac{1}{r-1}}$ times, corresponding to the size of the constructed independent
 1001 set. □

1002 **Theorem 17** *For every $r \geq 2$, MIS in $(K_r \setminus K_{1,r-2})$ -free graphs has a poly-*
 1003 *nomial Turing kernel.*

1004 *Proof.* The problem is polynomial for $r = 2$ and $r = 3$, hence we suppose
 1005 $r \geq 4$. Suppose we have an algorithm \mathcal{A} which, given a graph J and an integer
 1006 i such that $|V(J)| = O(i^{r-1})$, decides whether J has an independent set of

⁸ The non-degree of a vertex is the size of its non-neighborhood.

size i in constant time. Having a polynomial algorithm for MIS assuming the existence of \mathcal{A} implies a polynomial Turing kernel for the problem [10]. To do so, we will present an algorithm \mathcal{B} which, given a *connected* graph G and an integer k , outputs a polynomial (in $|V(G)|$) number of instances of size $O(k^{r-1})$, such that one of them is positive iff the former one is. With this algorithm in hand, we obtain the polynomial Turing kernel as follows: let G and k be an instance of MIS. Let V_1, \dots, V_ℓ be the connected components of G . For every $j \in \{1, \dots, \ell\}$, we determine the size of a maximum independent set k_j of $G[V_j]$ by first invoking, for successive values $i = 1, \dots, k$, the algorithm \mathcal{B} on input $(G[V_j], i)$, and then \mathcal{A} on each reduced instance. At the end of the algorithm, we answer *YES* iff $\sum_{j=1}^{\ell} k_j \geq k$.

We now describe the algorithm \mathcal{B} . Let (G, k) be an input, with $n = |V(G)|$. We first invoke Lemma 4. If the algorithm outputs an independent set of size at least $s = n^{\frac{1}{r-1}}$, then either $k \leq s$ and we are done (we output a trivially positive instance), or $k > n^{\frac{1}{r-1}}$ which implies that the instance is a kernel with $O(k^{r-1})$ vertices. Hence, we assume that the algorithm outputs a clique C of size at least $n^{\frac{1}{r-1}}$. We assume that $|C| > r^2$, since otherwise the instance is already reduced.

Let $B = N(C)$. First observe that for every $u \in B$, $|N_C(u)| \geq |C| - (r-3)$. Indeed, if $|N_C(u)| \leq |C| - (r-2)$, then the graph induced by $r-2$ non-neighbors of u in C together with u and a neighbor of u in C (which exists since $|C| > r^2$) is isomorphic to $K_r \setminus K_{1,r-2}$. Secondly, we claim that $V(G) = C \cup B$: for the sake of contradiction, take $v \in N(B) \setminus C$, and let $u \in B$ be such that $uv \in E(G)$. By the previous argument, u has at least $|C| - r + 3 \geq r - 2$ neighbors in C which, in addition to u and v , induce a graph isomorphic to $K_r \setminus K_{1,r-2}$.

The algorithm outputs, for every $u \in B$, the graph induced by $B \setminus N[u]$ (with parameter $k-1$), and, for every $u \in B$ and every $v \in C$ such that $uv \notin E(G)$, the graph induced by $B \setminus (N[u] \cup N[v])$ (with parameter $k-2$). The correctness of the algorithm follows from the fact that if G has an independent set S of size $k > 1$, then either:

- $S \cap C = \emptyset$, in which case $S \setminus \{u\}$ lies entirely in $B \setminus N[u]$ for any $u \in S$, or
- $S \cap C = \{v\}$ for some $v \in C$, in which case $S \setminus \{u, v\}$ lies entirely in $B \setminus (N[u] \cup N[v])$ for any $u \in S \cap B$.

We now argue that each of these instances has $O(k^{r-3})$ vertices. To do so, observe that for any $u \in B$, $B \setminus N[u]$ does not contain K_{r-2} as an induced subgraph: indeed, since $|C| > r^2$, then any set of $r-1$ vertices of B must have a common neighbor in C (since the union of the non-neighborhoods of these $r-1$ vertices in C is of size at most $(r-1)(r-3)$). Now, take (for the sake of contradiction) any clique K of size $r-2$ in $B \setminus N[u]$, and consider a common neighbor $x \in C$ of $K \cup \{u\}$. Then $K \cup \{u, x\}$ induces a graph isomorphic to $K_r \setminus K_{1,r-2}$, which is impossible. Since each of these instances is K_{r-2} -free, applying Ramsey's theorem to each of them allows us to either construct an independent set of size $k-1$ in one of them (and thus output an independent set of size k in G), or to prove that each of them has at most

1052 $O(k^{r-3})$ vertices. At the end, this algorithm outputs $O(n^2)$ instances, each
 1053 having $O(k^{r-3})$ vertices. \square

1054 Since a $(K_r \setminus K_{1,r-1})$ -free graph is $(K_{r'} \setminus K_{1,r'-2})$ -free for $r' = r + 1$, we
 1055 have the following:

1056 **Corollary 2** *For every $r \geq 2$, MIS in $(K_r \setminus K_{1,r-1})$ -free graphs has a poly-*
 1057 *nomial Turing Kernel.*

1058 In other words, $(K_r \setminus K_{1,r-1})$ is a clique of size $r - 1$ plus an isolated
 1059 vertex. Observe that the previous corollary can actually be proved in a very
 1060 simple way: informally, we can “guess” a vertex v of the solution, and return its
 1061 non-neighborhood together with parameter $k - 1$. Since this non-neighborhood
 1062 is K_{r-1} -free, it can be reduced to a $O(k^{r-2})$ -sized instance. This is perhaps
 1063 the most simple example of a problem admitting a polynomial Turing kernel
 1064 but no polynomial kernel, unless $NP \subseteq coNP/poly$ (as we will prove later in
 1065 Theorem 19). By considering the complement of graphs, it implies the follow-
 1066 ing even simpler observation: MAXIMUM CLIQUE has a $O(k^2)$ Turing kernel
 1067 on *claw*-free graphs, but no polynomial kernel, under the same complexity-
 1068 theoretic assumption.

1069 **Theorem 18** *For every $r \geq 3$, MIS in $(K_r \setminus K_{1,2})$ -free graphs has a kernel*
 1070 *with $O(k^{r-1})$ vertices.*

1071 *Proof.* For $r = 3$, the problem is polynomial, so we assume $r \geq 4$. We first
 1072 invoke Lemma 4. If the algorithm outputs an independent set of size at least
 1073 $s = n^{\frac{1}{r-1}}$, then either $k \leq s$ and we are done (we output a trivially positive
 1074 instance), or $k > n^{\frac{1}{r-1}}$ which implies that the instance is a kernel with $O(k^{r-1})$
 1075 vertices. Hence, we assume that the algorithm outputs a clique C of size at
 1076 least $n^{\frac{1}{r-1}}$. We assume that this clique is maximal. We present a reduction
 1077 rule in the case $|C| > (k - 1)(r - 4) + 1$. If this rule cannot apply, then it
 1078 means that the number of vertices of the reduced instance is $O(k^{r-1})$.

1079 First observe that for every $u \in N(C)$, then either $|N_C(u)| = |C| - 1$, or
 1080 $|N_C(u)| \leq r - 4$ (recall that $N_C(u) = N(u) \cap C$). Indeed, first observe that
 1081 $N_C(u) < |C|$, since C is maximal. Then, suppose that $r - 3 \leq |N_C(u)| \leq$
 1082 $|C| - 2$. Then u together with $r - 3$ of its neighbors in C and 2 of its non-
 1083 neighbors in C induce a graph isomorphic to $K_r \setminus K_{1,2}$, a contradiction. Let
 1084 $B = \{u \in N(C) : |N_C(u)| = |C| - 1\}$ and $D = \{u \in N(C) : |N_C(u)| \leq r - 4\}$.

1085 We claim that $C \cup B$ is a complete $|C|$ -partite graph. To do so, we prove
 1086 that for $u, v \in B$, $N_C(u) = N_C(v)$ implies $uv \notin E(G)$, and $N_C(u) \neq N_C(v)$
 1087 implies $uv \in E(G)$. Suppose that $N_C(u) = N_C(v) = C \setminus \{x\}$. If $uv \in E(G)$,
 1088 then u, v, x together with $r - 3$ vertices of C different from x induce a graph
 1089 isomorphic to $K_r \setminus K_{1,2}$, which is impossible. Suppose now that $N_C(u) =$
 1090 $C \setminus \{x_u\}$, $N_C(v) = C \setminus \{x_v\}$, with $x_u \neq x_v$. If $uv \notin E(G)$, then u, v, x_u
 1091 together with $r - 3$ vertices of C different from x_u and x_v induce a graph
 1092 isomorphic to $K_r \setminus K_{1,2}$, which is impossible.

1093 Thus, we now write $C \cup B = S_1 \cup \dots \cup S_{|C|}$, where, for every $i, j \in$
 1094 $\{1, \dots, |C|\}$, $i \neq j$, S_i induces an independent set, and $S_i \cup S_j$ induces a

complete bipartite graph. We assume $|S_1| \geq |S_2| \geq \dots \geq |S_{|C|}|$. Recall that $|C| > (k-1)(r-4) + 1$. Using the same arguments as previously, we can show that every vertex of D is adjacent to at most $r-4$ different parts among $C \cup B$: if a vertex $u \in D$ is adjacent to $r-3$ parts, then taking one vertex in each of these parts together with u and 2 non-neighbors of u in C induces a graph isomorphic to $K_r \setminus K_{1,2}$. Hence, for every $u \in D$, we have $|\{S_i : N(u) \cap S_i \neq \emptyset\}| \leq r-4$. Let $q = (k-1)(r-4) + 1$. The reduction consists of removing $S_{q+1} \cup \dots \cup S_{|C|}$. Clearly it runs in polynomial time.

Let G' denote the reduced instance. We now prove the safeness of this reduction rule. Obviously, if G' has an independent set of size k , then G does, since G' is an induced subgraph of G . It remains to show that the converse is also true. Let X be an independent set of G of size k . If $X \cap \left(\bigcup_{i=q+1}^{|C|} S_i\right) = \emptyset$, then X is also an independent set of size k in G' , thus we suppose $X \cap \left(\bigcup_{i=q+1}^{|C|} S_i\right) = X_r \neq \emptyset$, which implies that $|X \cap D| \leq k-1$. In particular, since $C \cup B$ is a complete multipartite graph, there is a unique $i \in \{1, \dots, |C|\}$ such that $X \cap S_i \neq \emptyset$, and $i \geq q+1$. Since every vertex of D is adjacent to at most $r-4$ parts of $C \cup B$, and since $q = (k-1)(r-4) + 1$, there must exist $j \in \{1, \dots, q\}$ such that $N(X \cap D) \cap S_j = \emptyset$. Moreover, $|S_j| \geq |S_i|$. Hence, $(X \setminus S_i) \cup S_j$ is an independent set of size at least k in G' .

Recall that we apply this reduction rule as long as $|C| > (k-1)(r-4) + 1$. If it is not the case, then the instance has $O(k^{r-1})$ vertices, since, by Lemma 4, we have $|C| \geq n^{\frac{1}{r-1}}$, and thus $n \leq (kr+5)^{r-1}$, which concludes the proof. \square

Observe that a $(K_r \setminus K_2)$ -free graph is $(K_{r+1} \setminus K_{1,2})$ -free, hence we have the following, which answers a question of [12].

Corollary 3 *For every $r \geq 1$, MIS in $(K_r \setminus K_2)$ -free graphs has a kernel with $O(k^{r-1})$ vertices.*

5.2 Kernel lower bounds

We now give a sufficient criteria for a graph H to preclude any polynomial kernel for MIS in H -free graphs. In a nutshell, we characterize graphs which cannot appear in the “straightforward” cross-composition consisting in taking the complete join of several instances.

Definition 8 Given a graph H , a *join* is a bipartition of $V(H)$ into two non-empty subsets (A, B) such that for every $a \in A$ and $b \in B$, $ab \in E(H)$.

Theorem 19 *Let H be any fixed graph such that (i) MIS is NP-hard in H -free graphs, and (ii) H has no join. Then MIS does not admit a polynomial kernel in H -free graphs unless $NP \subseteq \text{coNP}/\text{poly}$.*

Proof. We construct an OR-cross-composition from MIS in H -free graphs. For more details about cross-compositions, see [4]. Let G_1, \dots, G_t be a sequence of H -free graphs, and let $G' = G_1 + \dots + G_t$ (recall that $+$ is the join operation,

1134 that it, there are all possible edges between $V(G_i)$ and $V(G_j)$, $i \neq j$. Then
 1135 we have the following:

- 1136 – $\alpha(G') = \max_{i=1\dots t} \alpha(G_i)$, since, by construction of G' , any independent set
 1137 cannot intersect the vertex set of two distinct graphs G_i and G_j .
- 1138 – G' is H -free. Indeed, suppose that $X \subseteq V(G')$ induces a graph isomorphic
 1139 to H , and let $X_j = X \cap V(G_j)$ for every $j \in [t]$. Since every G_i is H -free,
 1140 at least two sets $X_j, X_{j'}$, $j \neq j'$ are non-empty. But then $(X_j, \cup_{s \neq j} X_s)$ is
 1141 a join in H , a contradiction.

1142 These two arguments imply a cross-composition from MIS in H -free graphs
 1143 to MIS in H -free graphs. \square

1144 Naturally, the previous lower bound also holds for graphs H containing
 1145 a graph H' as an induced subgraph fulfilling the statement of the theorem
 1146 (since the class of H' -free graphs is included in the class of H -free graphs).

1147 We now use this theorem to show that the polynomial kernel obtained in
 1148 the previous section for $(K_r \setminus K_{1,s})$ -free graphs, $s \leq 2$, is somehow tight.

1149 **Corollary 4** *For $r \geq 4$, and every $3 \leq s \leq r - 1$, MIS in $(K_r \setminus K_{1,s})$ -free
 1150 graphs does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

1151 *Proof.* Observe that for these values of r and s , $(K_r \setminus K_{1,s})$ always contain as
 1152 an induced subgraph the graph H defined as the disjoint union of K_1 and K_3 ,
 1153 which does not have a join, while MIS is NP-hard in H -free (since it contains
 1154 a triangle K_3). \square

1155 It would be interesting to find out whether there exist graphs H not falling
 1156 into the statement of Theorem 19 for which there is no polynomial kernel. In
 1157 other words: is Theorem 19 the only way to obtain kernel lower bounds in this
 1158 case?

1159 6 Conclusion and open problems

1160 We made some significant progress toward the FPT/ $W[1]$ -hard dichotomy for
 1161 MIS in H -free graphs, for a fixed graph H . At the cost of one reduction, we
 1162 showed that it is $W[1]$ -hard as soon as H is not chordal, even if we simulta-
 1163 neously forbid induced $K_{1,4}$ and trees with at least two branching vertices.
 1164 Tuning this construction, it is also possible to show that if a connected H is
 1165 not roughly a "path of cliques" or a "subdivided claw of cliques", then MIS
 1166 is $W[1]$ -hard. More formally, with the definitions of Section 2.2, the remaining
 1167 connected open cases are when H has an almost strong clique decomposition
 1168 on a subdivided claw or a nearly strong clique decomposition on a path. In this
 1169 language, we showed that for every connected graph H with a strong clique
 1170 decomposition on a P_3 , there is an FPT algorithm. However, we also proved
 1171 that for a very simple graph H with a strong clique decomposition on the claw,
 1172 MIS is $W[1]$ -hard. This suggests that the FPT/ $W[1]$ -hard dichotomy will be

1173 somewhat subtle. For instance, easy cases for the parameterized complexity
1174 do *not* coincide with easy cases for the classical complexity where each vertex
1175 can be blown into a clique. For graphs H with a clique decomposition on a
1176 path, the first unsolved cases are H having:

- 1177 – an almost strong clique decomposition on P_3 ;
- 1178 – a nearly strong clique decomposition on P_3 ;
- 1179 – a strong clique decomposition on P_4 .

1180 For graphs H with a clique decomposition on the claw, an interesting open
1181 question is the case of $T_{1,1,s}$ -free graphs (see notation preceding Theorem 4).
1182 We observe that a randomized FPT algorithm was later found in the $T_{1,1,2}$ -
1183 free (or *cricket*-free) case [5], while $W[1]$ -hardness on $T_{1,2,2}$ -free is established
1184 in this paper (see Theorem 4)

1185 For disconnected graphs H , we obtained an FPT algorithm when H is a
1186 cluster (*i.e.*, a disjoint union of cliques). We conjecture that, more generally,
1187 the disjoint union of two easy cases is an easy case; formally, *if MIS is FPT*
1188 *in G -free graphs and in H -free graphs, then it is FPT in $G \uplus H$ -free graphs.*

1189 A natural question regarding our two FPT algorithms of Section 4 concerns
1190 the existence of polynomial kernels. In particular, we even do not know whether
1191 the problem admits a kernel for very simple cases, such as when $H = K_5 \setminus K_3$
1192 or $H = K_5 \setminus K_{2,2}$.

1193 A more anecdotal conclusion is the fact that the parameterized complexity
1194 of the problem on H -free graphs is now complete for every graph H on four
1195 vertices, including concerning the polynomial kernel question (see Figure 9).
1196 Observe that the FPT/ $W[1]$ -hard dichotomy was recently settled for all graphs
1197 on five vertices [5], using tools from this paper.

1198 **Acknowledgements** N. B. and P. C. are supported by the ANR Project DISTANCIA
1199 (ANR-17-CE40-0015) operated by the French National Research Agency (ANR).

1200 We would like to thank two anonymous reviewers for their valuable comments which
1201 greatly improved the presentation of the paper.

1202 References

- 1203 1. Alekseev, V.E.: The effect of local constraints on the complexity of de-
1204 termination of the graph independence number. *Combinatorial-Algebraic*
1205 *Methods in Applied Mathematics* pp. 3–13 (1982). In Russian
- 1206 2. Alekseev, V.E.: On the number of maximal independent sets in graphs
1207 from hereditary classes. *Combinatorial-Algebraic Methods in Discrete Op-*
1208 *timization* pp. 5–8 (1991). (In Russian)
- 1209 3. Bacsó, G., Lokshtanov, D., Marx, D., Pilipczuk, M., Tuza, Z., van
1210 Leeuwen, E.J.: Subexponential-time algorithms for maximum indepen-
1211 dent set in P_t -free and broom-free graphs. *Algorithmica* **81**(2), 421–
1212 438 (2019). DOI 10.1007/s00453-018-0479-5. URL [https://doi.org/](https://doi.org/10.1007/s00453-018-0479-5)
1213 [10.1007/s00453-018-0479-5](https://doi.org/10.1007/s00453-018-0479-5)












Graph	P	PK	PTK	FPT
	Obvious			
	Obvious			
	Obvious			
	[2]			
	[22]			
	[8]			
	Thm. 1	Ramsey		
	Thm. 1	Cor. 3		
	Thm. 1	Thm. 18		
		Cor. 4	Cor. 2	
				Thm. 2

Fig. 9 Status of the problem for graphs H on four vertices. A green cell represents a positive answer while a red cell represents a negative answer under classical complexity assumptions. P , PK , PTK respectively stand for *Polynomial*, *NP-hard but admits a polynomial kernel*, and *no polynomial kernel unless $NP \subseteq coNP/poly$ but admits a polynomial Turing kernel*.

- 1214 4. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds
1215 by cross-composition. *SIAM J. Discrete Math.* **28**(1), 277–305 (2014)
- 1216 5. Bonnet, É., Bousquet, N., Thomassé, S., Watrigant, R.: When maxi-
1217 mum stable set can be solved in FPT time. In: P. Lu, G. Zhang (eds.)
1218 30th International Symposium on Algorithms and Computation, ISAAC
1219 2019, *LIPICs*, vol. 149, pp. 49:1–49:22. Schloss Dagstuhl - Leibniz-Zentrum
1220 für Informatik (2019). URL [https://doi.org/10.4230/LIPICs.ISAAC.](https://doi.org/10.4230/LIPICs.ISAAC.2019.49)
1221 2019.49
- 1222 6. Cai, L., Chan, S.M., Chan, S.O.: Random separation: A new method for
1223 solving fixed-cardinality optimization problems. In: Proceedings of the
1224 2nd International Workshop on Parameterized and Exact Computation
1225 (IWPEC), pp. 239–250 (2006)
- 1226 7. Chen, J., Liu, Y., Lu, S., Sze, S., Zhang, F.: Iterative expansion and color
1227 coding: An improved algorithm for 3d-matching. *ACM Trans. Algorithms*

- 1228 **8**(1), 6:1–6:22 (2012). DOI 10.1145/2071379.2071385. URL <https://doi.org/10.1145/2071379.2071385>
- 1229
- 1230 8. Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for
- 1231 cographs. *SIAM J. Comput.* **14**(4), 926–934 (1985)
- 1232 9. du Cray, H.P., Sau, I.: Improved FPT algorithms for weighted independent
- 1233 set in bull-free graphs. *Discrete Mathematics* **341**(2), 451–462 (2018)
- 1234 10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshтанov, D., Marx, D., Pilipczuk,
- 1235 M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer (2015)
- 1236 11. Dabrowski, K.: Structural solutions to maximum independent set and related
- 1237 problems. Ph.D. thesis, University of Warwick (2012)
- 1238 12. Dabrowski, K., Lozin, V.V., Müller, H., Rautenbach, D.: Parameterized
- 1239 complexity of the weighted independent set problem beyond graphs of
- 1240 bounded clique number. *J. Discrete Algorithms* **14**, 207–213 (2012)
- 1241 13. Diestel, R.: *Graph Theory*, 4th Edition, *Graduate texts in mathematics*,
- 1242 vol. 173. Springer (2012)
- 1243 14. Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity*.
- 1244 *Texts in Computer Science*. Springer (2013)
- 1245 15. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to*
- 1246 *the Theory of NP-Completeness*. W. H. Freeman (1979)
- 1247 16. Grzesik, A., Klimošová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time
- 1248 algorithm for maximum weight independent set on p_6 -free graphs. In:
- 1249 *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete*
- 1250 *Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*,
- 1251 pp. 1257–1271 (2019). DOI 10.1137/1.9781611975482.77. URL <https://doi.org/10.1137/1.9781611975482.77>
- 1252
- 1253 17. Hermelin, D., Mních, M., van Leeuwen, E.J.: Parameterized complexity of
- 1254 induced graph matching on claw-free graphs. *Algorithmica* **70**(3), 513–560
- 1255 (2014)
- 1256 18. Karthick, T.: Independent sets in some classes of $S_{i,j,k}$ -free graphs. *J.*
- 1257 *Comb. Optim.* **34**(2), 612–630 (2017)
- 1258 19. Karthick, T., Maffray, F.: Maximum weight independent sets in classes
- 1259 related to claw-free graphs. *Discrete Applied Mathematics* **216**, 233 – 239
- 1260 (2017)
- 1261 20. Lokshтанov, D., Vatshelle, M., Villanger, Y.: Independent set in P_5 -free
- 1262 graphs in polynomial time. In: *Proceedings of the Twenty-Fifth Annual*
- 1263 *ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pp. 570–581
- 1264 (2014)
- 1265 21. Marx, D.: Parameterized complexity of independence and domination on
- 1266 geometric graphs. In: *Parameterized and Exact Computation, Second*
- 1267 *International Workshop, IWPEC 2006, Zürich, Switzerland, September*
- 1268 *13-15, 2006, Proceedings*, pp. 154–165 (2006). DOI 10.1007/11847250_14.
- 1269 URL https://doi.org/10.1007/11847250_14
- 1270 22. Minty, G.J.: On maximal independent sets of vertices in claw-free graphs.
- 1271 *Journal of Combinatorial Theory, Series B* **28**(3), 284 – 304 (1980)
- 1272 23. Naor, M., Schulman, L.J., Srinivasan, A.: Splitters and near-optimal de-
- 1273 randomization. In: *Proceedings of IEEE 36th Annual Foundations of Com-*

- puter Science, pp. 182–191 (1995). DOI 10.1109/SFCS.1995.492475
- 1275 24. Poljak, S.: A note on stable sets and colorings of graphs. *Commentationes*
1276 *Mathematicae Universitatis Carolinae* pp. 307–309 (1974)
- 1277 25. Reed, B., Smith, K., Vetta, A.: Finding odd cycle transversals. *Operations*
1278 *Research Letters* **32**(4), 299 – 301 (2004)
- 1279 26. Sloper, C., Telle, J.A.: An overview of techniques for designing parameter-
1280 ized algorithms. *Comput. J.* **51**(1), 122–136 (2008). DOI 10.1093/comjnl/
1281 bxm038. URL <https://doi.org/10.1093/comjnl/bxm038>
- 1282 27. Thomassé, S., Trotignon, N., Vuskovic, K.: A Polynomial Turing-Kernel
1283 for Weighted Independent Set in Bull-Free Graphs. *Algorithmica* **77**(3),
1284 619–641 (2017)
- 1285 28. Zuckerman, D.: Linear degree extractors and the inapproximability of max
1286 clique and chromatic number. *Theory of Computing* **3**(1), 103–128 (2007)