Max Weight Independent Set in graphs with no long claws nor large induced bicliques

Maria Chudnovsky* Paweł Rzążewski†

Abstract

The complexity of Max Weight Independent Set (MWIS) in graphs excluding a fixed subdivided claw is one of the central open problems in algorithmic graph theory. We show that MWIS can be solved in polynomial time in graphs excluding a fixed subdivided claw and a fixed biclique as an induced subgraph. This strengthens a result of Abrishami, Chudnovsky, Dibek, Pilipczuk, and Rzążewski [ACM TALG 2026] who showed an analogous result, but under a stronger assumption that some biclique is excluded as a subgraph.

Contents

1 Introduction **1**

2 Preliminaries **2**

3 Auxiliary tools **4**
3.1 Hitting sets of small independence number 4
3.2 Obtaining a balanced extended strip decomposition 6
3.3 Small interfaces and component boundaries 7

4 Excluding a single long claw **9**
4.1 Auxiliary problem 9
4.2 Child instances 9
4.3 Proof of Theorem 1.1 for a single long claw 10

5 Excluding a disjoint union of long claws **11**

6 Conclusion **11**

*Princeton University. Supported by NSF Grant DMS-2348219, NSF Grant CCF-2505100, AFOSR grant FA9550-25-1-0275 and a Guggenheim Fellowship

†Warsaw University of Technology. Supported by the National Science Centre grant 2024/54/E/ST6/00094.

1 Introduction

Maximum Weight Independent Set (MWIS) is one of the fundamental problems in algorithmic graph theory. Given a graph with nonnegative vertex weights, the task is to find a set of pairwise nonadjacent vertices (i.e., an independent set) of maximum total weight. The unweighted version of this problem is equivalent, by graph complementation, to Clique, one of Karp's 21 NP-hard problems [19]. Nevertheless, MWIS admits polynomial-time algorithms in many important graph classes, including perfect graphs [16], claw-free graphs [13, 26, 29], and circle graphs (these are the intersection graphs of chords of a circle) [30].

We are interested in the complexity of MWIS in graph classes defined by forbidden induced subgraphs. For two graphs F and G, the graph G is *F-free* if G contains no induced subgraph isomorphic to F. More generally, G is *$(F_1, \dots, F_k)$-free* if it is F_i-free for every i. For $t \geq 1$, let P_t denote the path on t vertices, i.e., of length $t-1$. For $a, b, c \geq 1$, the *subdivided claw* $S_{a,b,c}$ is obtained from three paths of lengths a, b, c by identifying one endpoint of each path into a single vertex. In particular, $S_{1,1,1} = K_{1,3}$ is the three-leaf star, also known as the *claw*. Let $\mathcal{F}$ be the family of all graphs G such that each component of G is a subcubic tree with at most three leaves, or equivalently, a path or a subdivided claw.

Alekseev [4] showed that, for every fixed $F \notin \mathcal{F}$, even the problem of finding an independent set of maximum size (this is a special case of MWIS with all equal vertx weights) remains NP-hard in F-free graphs. For $F \in \mathcal{F}$, polynomial-time algorithms are known only in special cases. Notable examples include P_5-free graphs, by Lokshtanov, Vatshelle, and Villanger [22]; P_6-free graphs, by Grzesik, Klimošová, Pilipczuk, and Pilipczuk [17]; and fork-free graphs, by Lozin and Milanič [23], where the fork is $S_{2,1,1}$. There are also nontrivial tractable cases with disconnected forbidden graphs [6, 8].

Interestingly, for every $F \in \mathcal{F}$, MWIS in F-free graphs can be solved in *quasipolynomial* time. This was first established by Gartland and Lokshtanov [14] (see also [28]) for the case that F is a path, and then extended to general $F \in \mathcal{F}$ by Gartland, Lokshtanov, Masařík, Pilipczuk, Pilipczuk, and Rzążewski [15]. Whether these cases all admit polynomial-time algorithms remains wide open.

This motivates studying F-free graphs under additional restrictions. For example, polynomial-time algorithms for MWIS are known for graphs that exclude P_7 and some small graph [7,24,27]. Recently, MWIS was shown to be polynomial-time-solvable in P_7-free graphs of bounded clique number by Chudnovsky, Czyżewska, Kluk, Pilipczuk, and Rzążewski [9]. Another direction is to impose sparsity conditions. For every fixed $F \in \mathcal{F}$ and integer Δ, Abrishami, Chudnovsky, Dibek, and Rzążewski [3] gave a polynomial-time algorithm for MWIS in F-free graphs of maximum degree at most Δ. This was strengthened by Abrishami, Chudnovsky, Dibek, Pilipczuk, and Rzążewski [1], who replaced the degree bound with the assumption that a fixed biclique is excluded as a subgraph, not necessarily induced. In this paper we further generalize these results by requiring only that a fixed biclique be excluded as an induced subgraph.

Theorem 1.1. *For every fixed $s \geq 1$ and $F \in \mathcal{F}$, there is a polynomial-time algorithm which, given an $(F, K_{s,s})$-free graph G and a weight function $w : V(G) \to \mathbb{R}_{\geq 0}$, returns a maximum-weight independent set in (G, w).*

We emphasize that, by the celebrated Kövari–Sós–Turán theorem [20], excluding a fixed biclique as a subgraph forces the number of edges to be subquadratic, and in F-free graphs, the number of edges becomes even linear the result of Kuhn and Osthus [21]. Our setting permits quadratically many edges, so the algorithm must also accommodate dense graphs.

Our proof combines the algorithmic use of extended strip decompositions developed by Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [11] with the boundary formulation of Abrishami et al. [1]. An extended strip decomposition allows solutions on overlapping pieces, called *particles*, to be assembled in polynomial time. The key is to control the independence number of separators and boundaries, which may contain arbitrarily many vertices.

Bounding the independence number is sufficient for our purposes: every independent set contains only a bounded number of vertices from a set of bounded independence number, so all possible intersections of a solution with such a set can be enumerated in polynomial time. Combining structural results of Hajebi and Spirkl [18] with those of Bourneuf, Masaříková, Nadara, and Pilipczuk [5], we

show how to efficiently find, in a graph excluding a fixed subdivided claw and a fixed biclique, a set of bounded independence number whose deletion leaves a graph with an extended strip decomposition in which every particle contains at most half of the vertices from a prescribed set. We use such decompositions recursively: from each particle, we remove a set of bounded independence number and recurse on the components of the remaining graph. An appropriate choice of the prescribed set ensures that the recursion depth is logarithmic. For each such component, we consider every possible intersection of a solution with its *boundary*, which is a certain set of neighbors of the component, and compute the best way to extend it inside the component. A new ingredient is a bound on the independence number of these boundaries, which guarantees that only polynomially many such intersections need to be considered. As each vertex belongs to only a bounded number of the components considered in a single recursive call, the total running time is polynomial. Finally, if F has several components, we again use the machinery of Hajebi and Spirkl [18] to find a set of bounded independence number that intersects every induced copy of a sufficiently long subdivided claw; deleting this set reduces the problem to graphs excluding a single subdivided claw.

2 Preliminaries

Weights. Let G be a graph and let $\mathsf{w} : V(G) \to \mathbb{R}_{\geq 0}$ be a weight function on the vertices of G. A vertex-weighted graph is a pair (G, w). We also use w for its restriction to any induced subgraph of G. For a set $X \subseteq V(G)$, we define $\mathsf{w}(X) := \sum_{v \in X} \mathsf{w}(v)$, and we write $\mathsf{w}(G) := \mathsf{w}(V(G))$. Arithmetic operations and comparisons of weights take constant time.

In the paper we will use two types of weight functions: one coming from the input of MWIS, and another indicating some "important" vertices in the graph. To avoid confusion, the second function will be called a *mass function* and typically denoted by μ. Usually, the mass will be defined for a subset $U \subseteq V(G)$ as $\mu(v) = 1$ if $v \in U$ and $\mu(v) = 0$ otherwise. We will call such a function *indicator mass function of* U.

Graph theory. For two graphs G_1 and G_2, we denote by $G_1 + G_2$ the disjoint union of G_1 and G_2. For $X \subseteq V(G)$, let $N_G(X)$ be the set of neighbors of X, i.e., vertices outside X with a neighbor in X, and write $N_G[X] = X \cup N_G(X)$. Two sets are *anticomplete* if they are disjoint and no edge joins them. For a graph G, by $\alpha(G)$ we denote the independence number of G, i.e., the size of a maximum independent set in G. For a vertex-weighted graph (G, w), by $\alpha(G, \mathsf{w})$ we denote the maximum weight of an independent set in G with respect to w. If G is clear from the context, for $X \subseteq V(G)$ we will write $\alpha(X)$ and $\alpha(X, \mathsf{w})$ instead of $\alpha(G[X])$ and $\alpha(G[X], \mathsf{w})$, respectively.

Paths in graphs. Let G be a graph and let A, B be disjoint subsets of $V(G)$. By A–B-path we mean a path in G with one endpoint in A, the other endpoint in B, and all internal vertices in $V(G) \setminus (A \cup B)$. An *$A$–$B$-separator* is a set $Y \subseteq V(G) \setminus (A \cup B)$ that intersects all A–B-paths in G.

Extended strip decompositions. We follow the notation of [15, 25]. A *triangle* in a graph is a set of three distinct, pairwise adjacent vertices. For a graph H, by $T(H)$ we denote the set of triangles in H. We write xyz for a triangle $\{x, y, z\}$. An *extended strip decomposition* of a graph G is a pair (H, η) that consists of:

- a simple graph H called the *host*,
- a *vertex atom* $\eta(x) \subseteq V(G)$ for every $x \in V(H)$,
- an *edge atom* $\eta(xy) \subseteq V(G)$ for every $xy \in E(H)$, and its subsets $\eta(xy, x), \eta(xy, y) \subseteq \eta(xy)$ called *interfaces*, and
- a *triangle atom* $\eta(xyz) \subseteq V(G)$ for every $xyz \in T(H)$,

which have the following properties (empty sets are allowed):

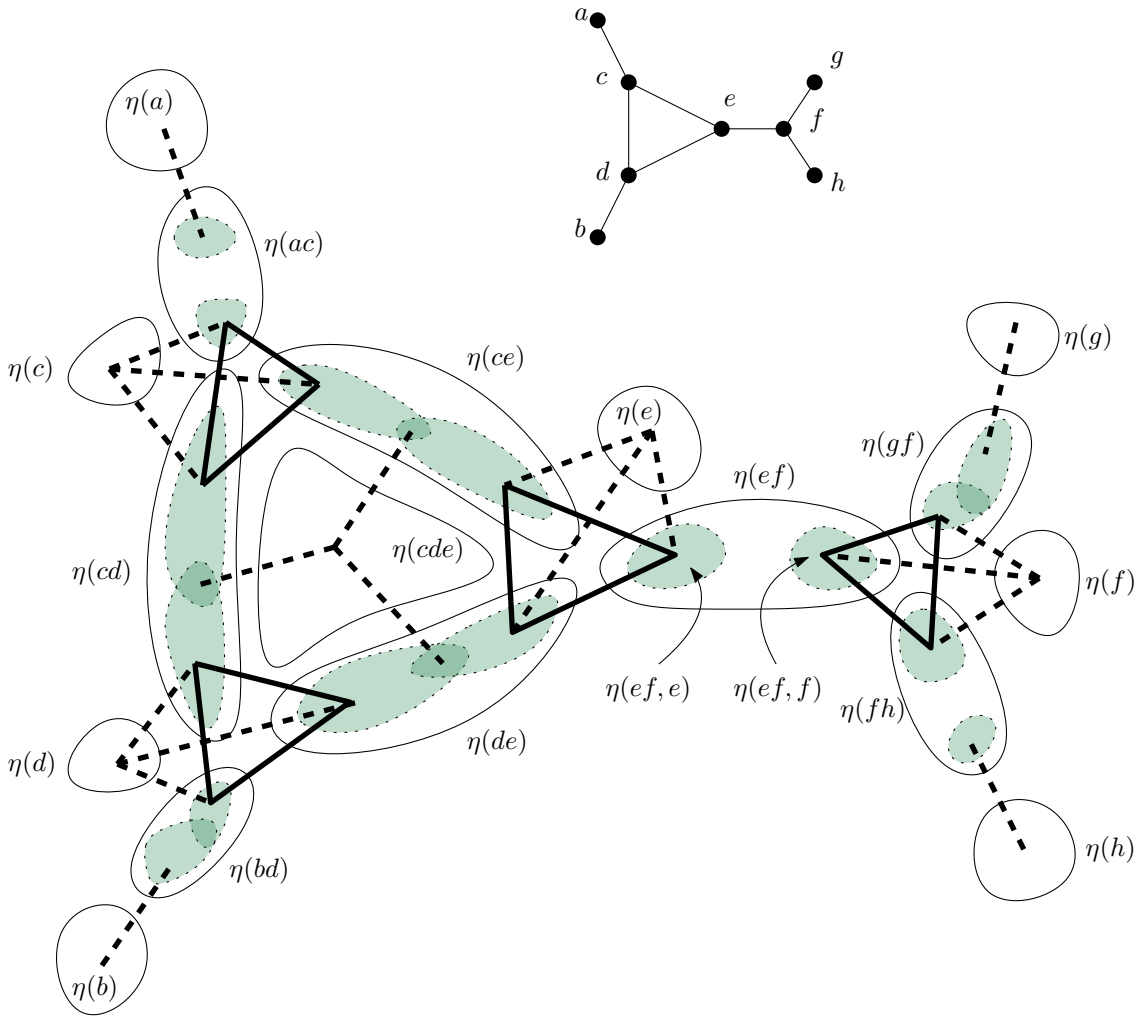


Figure 1: A graph H (top) and a schematic view of an extended strip decomposition (H, η) of a graph G. Green areas depict the sets $\eta(xy, y)$. Solid line indicates that all edges between two sets exist. Dashed line denotes that edges might exist. Picture from [15], courtesy of the authors.

1. The family $\{\eta(o) \mid o \in V(H) \cup E(H) \cup T(H)\}$ is a partition of $V(G)$.
2. For every $x \in V(H)$ and every distinct $y, z \in N_H(x)$, the set $\eta(xy, x)$ is complete to $\eta(xz, x)$ (i.e., all possible edges between these two sets exist).
3. Every $uv \in E(G)$ is contained in one of the atoms $\eta(o)$ for $o \in V(H) \cup E(H) \cup T(H)$, or is as follows:
   - $u \in \eta(xy, x), v \in \eta(xz, x)$ for some $x \in V(H)$ and $y, z \in N_H(x)$, or
   - $u \in \eta(xy, x), v \in \eta(x)$ for some $xy \in E(H)$, or
   - $u \in \eta(xyz)$ and $v \in \eta(xy, x) \cap \eta(xy, y)$ for some $xyz \in T(H)$.

A schematic picture of an extended strip decomposition of a graph is shown in Figure 1.

An extended strip decomposition (H, η) is *rigid* if for every $xy \in E(H)$, the sets $\eta(xy, x)$ and $\eta(xy, y)$ are nonempty, and for every isolated $x \in V(H)$, the vertex atom $\eta(x)$ is nonempty.

For an extended strip decomposition (H, η) of a graph G, we identify five *types* of *particles*.

$$\begin{aligned}
\text{vertex particle:} &\quad A_x := \eta(x) \text{ for each } x \in V(H),\\
\text{edge interior particle:} &\quad A^{\perp}_{xy} := \eta(xy) \setminus (\eta(xy,x) \cup \eta(xy,y)) \text{ for each } xy \in E(H),\\
\text{half-edge particle:} &\quad A^{x}_{xy} := \eta(x) \cup \eta(xy) \setminus \eta(xy,y) \text{ for each } xy \in E(H),\\
\text{full edge particle:} &\quad A^{xy}_{xy} := \eta(x) \cup \eta(y) \cup \eta(xy) \cup \bigcup_{z \mid xyz \in T(H)} \eta(xyz) \text{ for each } xy \in E(H),\\
\text{triangle particle:} &\quad A_{xyz} := \eta(xyz) \text{ for each } xyz \in T(H).
\end{aligned}$$

Given a mass function μ on $V(G)$, a decomposition of an induced subgraph of G is *balanced for* μ if every particle A satisfies $\mu(A) \leq \mu(G)/2$.

Let us conclude with listing some straightforward properties of extended strip decompositions that will be used in the paper. The first two follow directly from the definition of extended strip decompositions, and the last one is shown in [25, Observation 5].

Proposition 2.1. *The following hold.*

(1) Let G be a graph, let (H, η) be an extended strip decomposition of G, and let $X \subseteq V(G)$. By restricting η to X, we obtain an extended strip decomposition of $G[X]$, with every particle replaced by its intersection with X.

(2) Let G_1, G_2 be two graphs on disjoint vertex sets, let (H_1, η_1) and (H_2, η_2) be extended strip decompositions of G_1 and G_2, respectively. Then, by taking the disjoint union of H_1 and H_2 and the union of η_1 and η_2, we obtain an extended strip decomposition of $G_1 + G_2$.

(3) If (H, η) is a rigid extended strip decomposition of G, then $|V(H)| \leq 2|V(G)|$ and $|E(H)| \leq |V(G)|$.

The main algorithmic application of extended strip decompositions is that, given an extended strip decomposition (H, η) of (G, w), one can reduce computing $\alpha(G, \mathsf{w})$ to solving MWIS in particles of (H, η).

Theorem 2.2 (Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [11, Section 3.3])**.** *Given a weighted graph (G, w), an extended strip decomposition (H, η) of G, and the value $\alpha(A, \mathsf{w})$ for every particle A, one can compute $\alpha(G, \mathsf{w})$ in time polynomial in $|V(G)| + |V(H)|$.*

3 Auxiliary tools

In this section we develop a number of tools that will be used in the proof of Theorem 1.1.

3.1 Hitting sets of small independence number

For a graph G and an integer b, a *b-system in G* is a family of nonempty vertex sets of G, each of size at most b; the family itself may be empty. A set *hits* a family if it intersects every member. Note that an empty set system is hit by the empty set. Hajebi and Spirkl [18] proved that in graphs excluding a fixed induced biclique, b-systems admit a certain Erdős–Pósa-type property: either we can find many pairwise anticomplete members of the system, or a hitting set of small independence number.

Lemma 3.1 (Hajebi and Spirkl [18, Lemma 6.2])**.** *For $b, s, q \geq 1$, there is a computable constant $c_{3.1} = c_{3.1}(b, s, q) \in \mathbb{N}$ such that, for every $K_{s,s}$-free graph G and every b-system $\mathcal{S}$ in G, one of the following holds:*

1. *there is a hitting set $X \subseteq V(G)$ for $\mathcal{S}$ with $\alpha(X) \leq c_{3.1}$; or*
2. *there are q pairwise anticomplete members of $\mathcal{S}$.*

Lemma 3.2 (Hajebi and Spirkl [18, Theorem 3.1]). *For every $b, s, t \geq 1$, there are computable constants $c_{3.2} = c_{3.2}(b,s,t) \in \mathbb{N}$ and $d_{3.2} = d_{3.2}(b,s,t) \in \mathbb{N}$ such that, for every $(S_{t,t,t}, K_{s,s})$-free graph G and every b-system $\mathcal{S}$ in G, one of the following holds:*

1. *there is a hitting set $X \subseteq V(G)$ for $\mathcal{S}$ with $\alpha(X) \leq c_{3.2}$; or*
2. *there are anticomplete $S_1, S_2 \in \mathcal{S}$ and an S_1–S_2-separator Y with $\alpha(Y) \leq d_{3.2}$.*

Lemma 3.3 (Algorithmic Lemma 3.1). *For fixed $b, s, q \geq 1$, there is a computable constant $c_{3.3} = c_{3.3}(b,s,q)$ and a polynomial-time algorithm which, given a b-system $\mathcal{S}$ in a $K_{s,s}$-free graph G, returns either:*

1. *a hitting set X for $\mathcal{S}$ with $\alpha(X) \leq c_{3.3}$; or*
2. *q pairwise anticomplete members of $\mathcal{S}$.*

Proof. Enumerate all q-tuples of members of $\mathcal{S}$ and return an anticomplete tuple if it exists. This can be done in polynomial time, as both b and q are fixed.

Otherwise, Lemma 3.1 guarantees a hitting set of independence number at most $c_{3.1} = c_{3.1}(b,s,q)$. Without loss of generality we may assume that this hitting set is contained in $\bigcup \mathcal{S}$. Solve the rational feasibility program

$$\begin{aligned} 0 \leq x_v \leq 1 \qquad & \text{for every } v \in \bigcup \mathcal{S}, \\ \sum_{v \in S} x_v \geq 1 \qquad & \text{for every } S \in \mathcal{S}, \\ \sum_{v \in I} x_v \leq c_{3.1} \qquad & \text{for every independent set } I \subseteq \bigcup \mathcal{S} \text{ with } |I| = b \cdot c_{3.1} + 1. \end{aligned}$$

This linear program has a solution: indeed, the indicator vector of the guaranteed hitting set satisfies every constraint. Take $X = \{v \mid x_v \geq 1/b\}$.

By the second family of constraints, for every $S \in \mathcal{S}$, we have $\sum_{v \in S} x_v \geq 1$, and hence $X \cap S \neq \emptyset$. On the other hand, $\alpha(X) \leq b \cdot c_{3.1}$, as otherwise we would have an independent set $I \subseteq X \subseteq \bigcup \mathcal{S}$ of size $b \cdot c_{3.1} + 1$, and then $\sum_{v \in I} x_v \geq |I|/b > c_{3.1}$, violating the last constraint of the program. Thus we can set $c_{3.3}(b,s,q) = b \cdot c_{3.1}(b,s,q)$.

Note that the number of constraints is polynomial in $|V(G)|$, as b and $c_{3.1}$ are constants. As linear programs can be solved in polynomial time, the proof is complete. □

Lemma 3.4 (Algorithmic Lemma 3.2). *For every $b, s, t \geq 1$, there are constants $c_{3.4} = c_{3.4}(b,s,t) \in \mathbb{N}$ and $d_{3.4} = d_{3.4}(b,s,t) \in \mathbb{N}$ and a polynomial-time algorithm that, given an $(S_{t,t,t}, K_{s,s})$-free graph G and a b-system $\mathcal{S}$ in G, returns either:*

1. *a hitting set $X \subseteq V(G)$ for $\mathcal{S}$ with $\alpha(X) \leq c_{3.4}$; or*
2. *two anticomplete members $S_1, S_2 \in \mathcal{S}$ and an S_1–S_2-separator Y with $\alpha(Y) \leq d_{3.4}$.*

Proof. Define the following constants depending only on b, s, t:

$$\begin{aligned} q &= c_{3.2}(b,s,t) + 1, \\ c_{3.4} &= c_{3.3}(b,s,q), \\ d_{3.4} &= c_{3.3}(t, s, d_{3.2} + 2b + 1). \end{aligned}$$

Apply Lemma 3.3 to $\mathcal{S}$ with parameters b, s, q. If the call returns the first outcome, i.e., a hitting set with independence number at most $c_{3.4}$, we return it as the first outcome.

Thus, suppose that the application of Lemma 3.3 returns a subfamily $\mathcal{A} \subseteq \mathcal{S}$ consisting of q pairwise anticomplete members of $\mathcal{S}$. Every hitting set for $\mathcal{A}$ has independence number at least q, as the q sets in $\mathcal{A}$ are pairwise anticomplete and we have to pick a vertex from each of them. Thus Lemma 3.2 applied to $\mathcal{A}$ guarantees the existence of two members $\widehat{S}_1, \widehat{S}_2$ with a corresponding separator $\widehat{Y}$ of independence number at most $d_{3.2}$. We are left with finding such a pair and the corresponding separator of independence number at most $d_{3.4}$ in polynomial time.

For every unordered pair of distinct $S_1, S_2 \in \mathcal{A}$, we proceed as follows. Define $W = V(G) \setminus (S_1 \cup S_2)$ and construct a t-system $\mathcal{H}_{S_1,S_2}$ in $G[W]$ consisting of:

1. the sets of internal vertices of all induced S_1–S_2-paths of length at most t; and
2. all sets $\{v_1, \ldots, v_t\} \subseteq W$ such that $v_0, v_1, \ldots, v_t$ is an induced path for some $v_0 \in S_1$.

According to the definition above, we distinguish the members of $\mathcal{H}_{S_1,S_2}$ of the first and second type, respectively. Note that every S_1–S_2-path has at least one internal vertex, as S_1 and S_2 are anticomplete. Consequently, every member of $\mathcal{H}_{S_1,S_2}$ is nonempty and has size at most t.

Claim 3.4.1. *Every hitting set $Y \subseteq W$ for $\mathcal{H}_{S_1,S_2}$ is an S_1–S_2-separator.*

Proof of Claim 3.4.1: Suppose otherwise and let P be a shortest S_1–S_2-path in $G - Y$. In particular, P is induced. If P has length at most t, then the set of its internal vertices is a member of $\mathcal{H}_{S_1,S_2}$ of the first type, contradicting the choice of Y. Otherwise, t consecutive vertices of P, starting at the neighbor of the endpoint in S_1, form a member of $\mathcal{H}_{S_1,S_2}$ of the second type, again contradicting the choice of Y. ⌟

Claim 3.4.2. *For every S_1, S_2, the system $\mathcal{H}_{S_1,S_2}$ contains at most $2b$ pairwise anticomplete members of the second type.*

Proof of Claim 3.4.2: Suppose otherwise. By the pigeonhole principle, three witnessing paths would have a common endpoint in S_1, since $|S_1| \leq b$. As, except for the common endpoint, the paths are pairwise anticomplete, they would form an induced $S_{t,t,t}$ in G, a contradiction. ⌟

For the pair $\widehat{S}_1, \widehat{S}_2$, there are also at most $d_{3.2}$ pairwise anticomplete members of the first type: the separator $\widehat{Y}$ hits each of them, and choosing one vertex from each intersection gives an independent set in the separator. Consequently, $\mathcal{H}_{\widehat{S}_1,\widehat{S}_2}$ has no $d_{3.2} + 2b + 1$ pairwise anticomplete members.

Apply Lemma 3.3 to each $\mathcal{H}_{S_1,S_2}$ in $G[W]$ with parameters $t, s, d_{3.2} + 2b + 1$. Since the second outcome is impossible for the pair $\widehat{S}_1, \widehat{S}_2$, at least one call returns a hitting set Y for some $\mathcal{H}_{S_1,S_2}$ with $\alpha(Y) \leq d_{3.4}$. As Lemma 3.3 was invoked in $G[W]$, we have that $Y \subseteq W$. Thus, by Claim 3.4.1, Y is an S_1–S_2-separator, and we can return S_1, S_2, Y as the second outcome of Lemma 3.4.

Note that there are $\binom{q}{2}$ pairs S_1, S_2 and each family can be constructed by exhaustively enumerating paths of length at most t. As all parameters are fixed, the running time is polynomial. □

3.2 Obtaining a balanced extended strip decomposition

Bourneuf, Masaříková, Nadara, and Pilipczuk [5] proved that, given an $S_{t,t,t}$-free graph and a mass function, one can remove a constant number of closed neighborhoods to obtain a rigid balanced extended strip decomposition of the remaining graph.

Theorem 3.5 ([5, Theorem 3])**.** *Given an $S_{t,t,t}$-free graph G and a mass function μ, one can in polynomial time find $S \subseteq V(G)$ with $|S| \leq 3t + 11$ and a rigid extended strip decomposition of $G - N_G[S]$ that is balanced for μ.*

In this subsection we aim to obtain an analogous statement, but instead of removing few closed neighborhoods, we will remove a set of vertices of bounded independence number. We remark that an existential statement of Lemma 3.6 can be quite easily obtained by combining Theorem 3.5 with a result of Hajebi and Spirkl [18, Lemma 7.2]. However, we need an algorithmic variant.

Lemma 3.6. *For fixed s, t, there is a computable constant $a_{3.6}$ and a polynomial-time algorithm which, given an n-vertex $(S_{t,t,t}, K_{s,s})$-free graph G with mass function μ, finds Z with $\alpha(Z) \leq a_{3.6}$ and an extended strip decomposition of $G - Z$ that is balanced for μ and whose host has $\mathcal{O}(n)$ vertices and edges.*

Proof. Define the following constants depending only on s, t:

$$\begin{aligned} b &= 3t + 11, \\ c &= c_{3.4}(b, s, t), \\ d &= d_{3.4}(b, s, t), \\ a_{3.6} &= 2c + d. \end{aligned}$$

Throughout the execution of the algorithm, we will maintain a b-system $\mathcal{S}$; the algorithm starts with $\mathcal{S} = \emptyset$. With each $S \in \mathcal{S}$ we store a set X_S with $\alpha(X_S) \le c$ and a rigid extended strip decomposition (H_S, η_S) of $G - (X_S \cup N_G[S])$ that is balanced for μ. By Proposition 2.1(3), every stored host has at most $2n$ vertices and n edges.

Iteratively, we apply Lemma 3.4 to the current system $\mathcal{S}$.

Suppose first that the application returns the second outcome, i.e., two anticomplete members $S_1, S_2 \in \mathcal{S}$ and an S_1–S_2-separator Y with $\alpha(Y) \le d$. Define $Z = X_{S_1} \cup X_{S_2} \cup Y$, so $\alpha(Z) \le 2c+d = a_{3.6}$. No component of $G - Z$ meets both $N_G[S_1]$ and $N_G[S_2]$: such a component, together with an endpoint in each S_i, would give an S_1–S_2-path in $G - Y$. Let V_1 be the union of the components avoiding $N_G[S_1]$, and set $V_2 = V(G - Z) \setminus V_1$. Then V_1, V_2 are anticomplete and

$$V_i \subseteq V(G - (X_{S_i} \cup N_G[S_i])) \qquad \text{for } i \in \{1, 2\}.$$

Restrict (H_{S_i}, η_{S_i}) to V_i and take the disjoint union of the two decompositions. By Proposition 2.1(1) and (2), this is a decomposition of $G - Z$, and each new particle is contained in a stored particle. Hence the result is balanced for μ. Restriction leaves the hosts unchanged, so their union has at most $4n$ vertices and at most $2n$ edges. Return Z and this decomposition, and terminate. We remark that the restrictions may destroy rigidity, but the outcome of this step is never stored in $\mathcal{S}$.

So, suppose that the application of Lemma 3.4 returns the first outcome, i.e., a hitting set X for $\mathcal{S}$ with $\alpha(X) \le c$. Apply Theorem 3.5 to $G - X$ with the restriction of μ. We obtain a set $S \subseteq V(G) \setminus X$ with $|S| \le b$ and a rigid extended strip decomposition (H, η) of $(G - X) - N_{G-X}[S] = G - (X \cup N_G[S])$ whose particles have mass at most $\mu(G - X)/2 \le \mu(G)/2$. If $S = \emptyset$, return $Z = X$ and (H, η), and terminate. This satisfies the required independence-number bound since $\alpha(X) \le c \le a_{3.6}$, and the host has at most $2n$ vertices and n edges by rigidity.

Otherwise, add S to $\mathcal{S}$, set $X_S = X$ and $(H_S, \eta_S) = (H, \eta)$, and proceed to the next iteration. Note that S was not already a member of $\mathcal{S}$: it is disjoint from X, whereas every previous member meets X. Thus every nonterminating iteration adds a new member to $\mathcal{S}$, and there are at most n^b possible members. Each iteration takes polynomial time, using Lemma 3.4 and Theorem 3.5 and, in the terminating separator case, restriction and disjoint union. Hence the entire algorithm runs in polynomial time. □

3.3 Small interfaces and component boundaries

Fix $s \ge 2$ and let (H, η) be an extended strip decomposition of a $K_{s,s}$-free graph G. Call a vertex set *small* if its independence number is at most $s - 1$, and *large* otherwise. At most one interface at a given host vertex is large: two such interfaces would contain independent s-vertex sets complete to one another, inducing a $K_{s,s}$. For every $x \in V(H)$, define

$$\mathsf{small}(x) = \bigcup_{\substack{xy \in E(H) \\ \eta(xy,x) \text{ small}}} \eta(xy, x), \qquad \mathsf{Small} = \bigcup_{x \in V(H)} \mathsf{small}(x). \tag{1}$$

Thus Small is the union of all small interfaces. As all interfaces at x are complete to each other, we have $\alpha(\mathsf{small}(x)) \le s - 1$. However, the independence number of Small might be arbitrarily large. For fixed s, sets $\mathsf{small}(x)$ and Small can be found in polynomial time by exhaustively enumerating s-vertex sets in each interface.

Lemma 3.7. *Let $s \ge 2$ be an integer, let (H, η) be an extended strip decomposition of a $K_{s,s}$-free graph G. Let $\mathsf{small}(x)$ for each $x \in V(H)$ and Small be defined as in (1). Then the following hold.*

(1) $\alpha(\mathsf{Small} \cap A) \le 2(s - 1)$ for every particle A of (H, η).

(2) Every component D of $G - \mathsf{Small}$ is contained in a particle of (H, η) and satisfies $\alpha(N_G(D)) \le 2(s - 1)$.

Proof. The set Small lies entirely in the union of edge atoms, and, for every edge $xy \in E(H)$, $\mathsf{Small} \cap \eta(xy)$ is the union of the small interfaces of xy. Hence $\alpha(\mathsf{Small} \cap \eta(xy)) \le 2(s-1)$. Since every particle meets at most one edge atom, this proves item (1).

To prove item (2), let D be a component of $G - \mathsf{Small}$; in particular, $N_G(D) \subseteq \mathsf{Small}$.

Claim 3.7.1. *The component D is contained in a particle of (H, η).*

Proof of Claim 3.7.1: Let (H, η') be the extended strip decomposition of $G - \mathsf{Small}$ obtained by restricting (H, η) to $V(G) \setminus \mathsf{Small}$, as in Proposition 2.1(1). Every small interface becomes empty, so at each host vertex at most one interface of (H, η') is nonempty. By the definition of an extended strip decomposition, every edge between two distinct edge atoms joins two interfaces at a common host vertex. Hence distinct edge atoms of (H, η') are anticomplete. Similarly, all neighbors of a vertex atom $\eta'(x)$ lie in the unique nonempty interface at x, if it exists, so $\eta'(x)$ has neighbors in at most one edge atom. A triangle atom $\eta'(xyz)$ has neighbors only in the edge atoms $\eta'(xy)$, $\eta'(xz)$, and $\eta'(yz)$. Moreover, it has a neighbor in $\eta'(xy)$ only if both $\eta'(xy, x)$ and $\eta'(xy, y)$ are nonempty, and analogously for xz and yz. As any two of the edges xy, xz, and yz of H share a vertex, and at each host vertex at most one interface of (H, η') is nonempty, at most one of these three edges has both of its interfaces in (H, η') nonempty. Hence $\eta'(xyz)$ also has neighbors in at most one edge atom. Distinct vertex and triangle atoms are anticomplete. Consequently, as D is connected, it meets at most one edge atom of (H, η'). If D meets $\eta'(xy)$, then, as D is connected and meets no other edge atom, every vertex or triangle atom meeting D has a neighbor in $\eta'(xy)$. Thus every such atom is $\eta'(x)$, $\eta'(y)$, or $\eta'(xyz)$ for some $xyz \in T(H)$, and hence $D \subseteq A^{xy}_{xy}$. Otherwise, D meets no edge atom, so it is contained in a single vertex or triangle atom, as distinct vertex and triangle atoms are anticomplete. In both cases, D is contained in a particle of (H, η). ⌟

Thus, it remains to show that $\alpha(N_G(D)) \leq 2(s-1)$. Let us first consider the case when D meets an edge atom $\eta(xy)$ for some $xy \in E(H)$.

Claim 3.7.2. *If D meets $\eta(xy)$ for some $xy \in E(H)$, then $N_G(D) \subseteq \mathsf{small}(x) \cup \mathsf{small}(y)$. In particular, $\alpha(N_G(D)) \leq \alpha(\mathsf{small}(x)) + \alpha(\mathsf{small}(y)) \leq 2(s-1)$.*

Proof of Claim 3.7.2: Every particle meeting $\eta(xy)$ is one of $A^{\perp}_{xy}$, A^{x}_{xy}, A^{y}_{xy}, and A^{xy}_{xy}, and all of them are contained in A^{xy}_{xy}. Hence $D \subseteq A^{xy}_{xy}$ by Claim 3.7.1. Neighbors of D in $\eta(xy)$ lie in $\mathsf{Small} \cap \eta(xy)$, and thus in $\mathsf{small}(x) \cup \mathsf{small}(y)$. Since $N_G(D) \subseteq \mathsf{Small}$, every neighbor of D lies in an edge atom. By the definition of an extended strip decomposition, every neighbor of D outside $\eta(xy)$ thus lies in an interface $\eta(xz, x)$ with $z \in N_H(x) \setminus \{y\}$ or in an interface $\eta(yz, y)$ with $z \in N_H(y) \setminus \{x\}$. By symmetry, let w be a neighbor of D of the first kind; we show that $w \in \mathsf{small}(x)$. The only vertices of A^{xy}_{xy} that can be adjacent to w lie in $\eta(x)$, in $\eta(xy, x)$, or in $\eta(xyz)$ if $xyz \in T(H)$, so D meets one of these sets. In fact, D meets $\eta(xy, x)$. Indeed, all neighbors of $\eta(x)$ and of $\eta(xyz)$ in A^{xy}_{xy} lie in $\eta(xy, x)$, and if D meets one of these two atoms, then it also contains a neighbor of this atom, as D is connected and meets $\eta(xy)$. As $D \cap \mathsf{Small} = \emptyset$, the interface $\eta(xy, x)$ is not contained in Small, so it is large. Since at most one interface at x is large, $\eta(xz, x)$ is small, and hence $w \in \mathsf{small}(x)$. Now, the bound on $\alpha(N_G(D))$ follows from the bound on $\alpha(\mathsf{small}(x))$ and $\alpha(\mathsf{small}(y))$. ⌟

Now, consider the case when D meets no edge atom, i.e., it is contained in a single vertex or triangle atom. Since $N_G(D) \subseteq \mathsf{Small}$, the definition of an extended strip decomposition implies the following:

- if $D \subseteq \eta(x)$ for some $x \in V(H)$, then $N_G(D)$ is contained in the union of the sets $\mathsf{Small} \cap \eta(xy, x)$ over all $y \in N_H(x)$;
- if $D \subseteq \eta(xyz)$ for some $xyz \in T(H)$, then $N_G(D)$ is contained in the union of the sets $\mathsf{Small} \cap \eta(xy, x) \cap \eta(xy, y)$, $\mathsf{Small} \cap \eta(xz, x) \cap \eta(xz, z)$, and $\mathsf{Small} \cap \eta(yz, y) \cap \eta(yz, z)$.

In both cases, any two of these sets are contained in two distinct interfaces at a common host vertex, and hence they are complete to each other. Thus every independent subset of $N_G(D)$ is contained in one of these sets. Moreover, each of these sets is contained in $\mathsf{Small} \cap A$ for some full edge particle A, so its independence number is at most $2(s-1)$ by item (1). Therefore $\alpha(N_G(D)) \leq 2(s-1)$, which completes the proof of item (2). □

4 Excluding a single long claw

In this section we prove Theorem 1.1 for $F = S_{t,t,t}$, using Lemmas 3.6 and 3.7. We follow the recursive scheme of Abrishami, Chudnovsky, Dibek, Pilipczuk, and Rzążewski [1, Section 3], keeping terminal sets of bounded independence number.

Fix integers $t \geq 1, s \geq 2$; we treat them as fixed constants. Fix an ambient $(S_{t,t,t}, K_{s,s})$-free weighted graph (G, w), which will remain unchanged throughout the recursion.

4.1 Auxiliary problem

In the recursion, we solve a more general problem than the original one. An *instance* is a pair (V', T) of disjoint subsets of $V(G)$, and its graph is $G' = G[V' \cup T]$ with weights w. The vertices of T are called *terminals* and the set V' is called the *interior* of the instance.

The goal for an instance (V', T) is to compute the auxiliary table entries $\mathsf{Tab}[V', T, I]$ for all independent sets $I \subseteq T$. Each entry stores the largest weight that the interior can contribute when precisely the terminals in I are selected:

$$\mathsf{Tab}[V', T, I] = \max\{\mathsf{w}(J) \mid J \subseteq V',\ I \cup J \text{ is independent in } G\}.$$

In particular, $\mathsf{Tab}[V(G), \emptyset, \emptyset] = \alpha(G, \mathsf{w})$.

For a set $Z \subseteq V' \cup T$ and an independent set $J \subseteq T \cup Z$, define $W_J = V' \setminus (Z \cup N_G(J))$. Then, for every independent $I \subseteq T$,

$$\mathsf{Tab}[V', T, I] = \max_{\substack{\text{independent } J \subseteq T \cup Z \\ J \cap T = I}} [\mathsf{w}(J \setminus T) + \alpha(G[W_J], \mathsf{w})]\,. \tag{2}$$

Indeed, after fixing the selected vertices J of $T \cup Z$, the remaining set is an independent set in $G[W_J]$.

4.2 Child instances

Fix an instance (V', T) with graph $G' = G[V' \cup T]$. We define child instances whose tables determine $\mathsf{Tab}[V', T, \cdot]$. Let $Z \subseteq V' \cup T$ and let (H, η) be an extended strip decomposition of $G' - Z$. Let Small be defined as in (1), and define $S = T \cup Z \cup \mathsf{Small}$. For every particle A, let $\mathcal{Q}(A)$ consist of the vertex sets of the components of $G[A \setminus S]$. Define $\mathcal{Q} = \bigcup_A \mathcal{Q}(A)$, where the union ranges over all particles of (H, η). For each $R \in \mathcal{Q}$, define its *boundary* $B_R = N_G(R) \cap S$ and the child instance (R, B_R), whose interior is R. Note that even when $R \in \mathcal{Q}(A)$, its boundary B_R need not be contained in A. We compute each child table once, even if R occurs for several particles.

Clearly, one can construct $\mathcal{Q}$ and all child instances in time polynomial in $|V' \cup T| + |V(H)|$.

Lemma 4.1. *Every vertex of V' belongs to at most four members of $\mathcal{Q}$. Consequently, we have*

$$\sum_{R \in \mathcal{Q}} |R| \leq 4|V'|.$$

Proof. Every particle is contained in $(V' \cup T) \setminus Z$ and $T \cup Z \subseteq S$, so every member of $\mathcal{Q}$ is a subset of $V' \setminus S$. Hence the second statement follows from the first, and it suffices to prove the first statement for a fixed vertex $v \in V' \setminus S$. For every particle A containing v, let C_A be the vertex set of the component of $G[A \setminus S]$ containing v. Then C_A is the unique member of $\mathcal{Q}(A)$ containing v, so the members of $\mathcal{Q}$ containing v are exactly the sets C_A, where A ranges over the particles containing v.

If v lies in an edge atom $\eta(xy)$, then every particle containing v is one of $A^{\perp}_{xy}$, A^{x}_{xy}, A^{y}_{xy}, and A^{xy}_{xy}. If v lies in a triangle atom $\eta(xyz)$, then every particle containing v is one of A_{xyz}, A^{xy}_{xy}, A^{xz}_{xz}, and A^{yz}_{yz}. In both cases, there are at most four sets C_A.

It remains to consider the case that v lies in a vertex atom $\eta(x)$ for some $x \in V(H)$. By the definition of particles, the particles containing v are exactly the following:

- the vertex particle A_x, and
- the half-edge particle A^{x}_{xy} and the full edge particle A^{xy}_{xy} for each $y \in N_H(x)$.

We show that at most three of the sets C_A are distinct.

By the definition of an extended strip decomposition, every vertex of $N_{G'-Z}(\eta(x))$ lies in an interface $\eta(xy,x)$ for some $y \in N_H(x)$. Let $y \in N_H(x)$ be such that $\eta(xy,x)$ is small, and let $A \in \{A^x_{xy}, A^{xy}_{xy}\}$. Since A is disjoint from all edge atoms other than $\eta(xy)$, we have $N_{G'-Z}(\eta(x)) \cap A \subseteq \eta(xy,x) \subseteq \mathsf{Small} \subseteq S$ by (1). Thus $\eta(x)$ is anticomplete to $A \setminus (S \cup \eta(x))$, and since C_A is connected and contains $v \in \eta(x)$, we obtain $C_A \subseteq \eta(x)$. As $A_x = \eta(x) \subseteq A$, this yields $C_A = C_{A_x}$.

Finally, recall that at most one interface at x is large. Hence every set C_A equals C_{A_x}, except possibly $C_{A^x_{xy}}$ and $C_{A^{xy}_{xy}}$ for the unique $y \in N_H(x)$ with $\eta(xy,x)$ large, if such a y exists. Thus at most three of the sets C_A are distinct. □

Lemma 4.2. *Given the child tables, we can compute* $\mathsf{Tab}[V', T, \cdot]$ *in time polynomial in* $|V' \cup T| + |V(H)|$ *per independent subset of* $T \cup Z$.

Proof. Fix an independent set $J \subseteq T \cup Z$ and let W_J be as in (2). For every particle A, we have

$$\alpha(A \cap W_J, \mathsf{w}) = \max_{\substack{I \subseteq \mathsf{Small} \cap A \cap W_J \\ I \text{ independent}}} \left[\mathsf{w}(I) + \sum_{R \in \mathcal{Q}(A)} \mathsf{Tab}[R, B_R, (I \cup J) \cap B_R] \right]. \tag{3}$$

Indeed, after fixing I, the remaining choices lie in the pairwise anticomplete components $R \in \mathcal{Q}(A)$. As $I \cup J \subseteq S$, the choices in each R must be compatible precisely with $(I \cup J) \cap B_R$, as encoded by the corresponding table entry. By Lemma 3.7(1), there are $|V' \cup T|^{\mathcal{O}(s)}$ choices of I. Restrict (H, η) to W_J using Proposition 2.1(1), and apply Theorem 2.2 with the values in (3) to compute $\alpha(G[W_J], \mathsf{w})$. Equation (2) gives the parent table. This uses only the supplied child tables. □

4.3 Proof of Theorem 1.1 for a single long claw

Theorem 4.3 (Theorem 1.1 for a single long claw)**.** *For every fixed* $s, t \geq 1$*, there is a polynomial-time algorithm which, given an* $(S_{t,t,t}, K_{s,s})$*-free graph* G *and a weight function* $\mathsf{w} : V(G) \to \mathbb{R}_{\geq 0}$*, returns a maximum-weight independent set in* (G, w).

Proof. Let $n = |V(G)|$. If $n \leq 1$ or $s = 1$, the result is obvious. Thus suppose $n \geq 2$ and $s \geq 2$, and define $k = a_{3.6} + 2(s-1)$; this constant depends only on s, t.

Throughout the recursion, we compute the table $\mathsf{Tab}[V', T, \cdot]$ for an instance (V', T) while maintaining $\alpha(T) \leq 6k$. The initial call uses $V' = V(G)$ and $T = \emptyset$. At each call, define $G' = G[V' \cup T]$. If $|V'| \leq 1$, compute each table entry directly. Otherwise proceed as follows.

1. Enumerate all independent subsets of T of size at most $6k$. Choose one of maximum cardinality, say J, and write $h = |J| = \alpha(T)$.

2. If $h \leq 5k$, let μ be the indicator mass function of V'. Otherwise, let μ be the indicator mass function of J. Apply Lemma 3.6 to G' with this mass function, obtaining Z with $\alpha(Z) \leq a_{3.6}$ and a balanced extended strip decomposition of $G' - Z$ with a host of linear size. Define its small-interface set Small as in (1).

3. Construct the distinct children (R, B_R) as in Section 4.2 and recursively compute $\mathsf{Tab}[R, B_R, \cdot]$, once per child.

4. Compute the parent table $\mathsf{Tab}[V', T, \cdot]$ using Lemma 4.2.

We verify that the terminal bound is preserved and that the recursion terminates. For a child (R, B_R), let D be the component of $G' - (Z \cup \mathsf{Small})$ containing R. Since $N_{G'}(D) \subseteq Z \cup N_{G'-Z}(D)$, Lemma 3.7(2) gives

$$\alpha(N_{G'}(D)) \leq k, \qquad B_R \subseteq (T \cap D) \cup N_{G'}(D).$$

If $h \leq 5k$, the decomposition is balanced for the indicator mass function of the interior. Since D is contained in a particle, we obtain

$$|R| \leq |D \setminus T| \leq |V'|/2, \qquad \alpha(B_R) \leq \alpha(T \cap D) + k \leq h + k \leq 6k.$$

Suppose instead that $h > 5k$. The decomposition is balanced for the indicator mass function of J, so $|J \cap D| \le h/2$. Replacing $J \cap N_{G'}[D]$ by any independent set in $T \cap D$ preserves independence, as $J \setminus N_{G'}[D]$ is anticomplete to D. Since J is a maximum independent set of T, this gives

$$\alpha(B_R) \le \alpha(T \cap D) + k \le |J \cap D| + |J \cap N_{G'}(D)| + k \le h/2 + 2k \le 5k.$$

Thus the terminal bound is preserved, and any call that does not halve the interior has children with terminal independence number at most $5k$. Every two recursion levels therefore halve the interior, so the recursion terminates. Each child graph is induced in G and hence remains $(S_{t,t,t}, K_{s,s})$-free. Correctness now follows by induction using Lemma 4.2. At the root, $\mathsf{Tab}[V(G), \emptyset, \emptyset] = \alpha(G, \mathsf{w})$.

The recursion has depth $\mathcal{O}(\log n)$, and Lemma 4.1 bounds the sum of the interior sizes at depth ℓ by $4^\ell n$. Since all child interiors are nonempty, the recursion tree has polynomially many nodes. At each node, Lemma 3.6 takes polynomial time and produces a host of linear size. The bounds $\alpha(T) \le 6k$ and $\alpha(T \cup Z) \le 7k$ give polynomially many table entries and assignments in Lemma 4.2. Hence each node takes polynomial time apart from its recursive calls, and the total running time is polynomial. Finally, a maximum-weight independent set can be recovered by standard bookkeeping. □

5 Excluding a disjoint union of long claws

In this section we prove Theorem 1.1. On a high level, another application of Lemma 3.3 reduces the problem to the case of a single long claw, which can then be solved using Theorem 4.3.

Theorem 1.1. *For every fixed $s \ge 1$ and $F \in \mathcal{F}$, there is a polynomial-time algorithm which, given an $(F, K_{s,s})$-free graph G and a weight function $\mathsf{w} : V(G) \to \mathbb{R}_{\ge 0}$, returns a maximum-weight independent set in (G, w).*

Proof. Let d be the number of components of F and let $t = \max\{1, |V(F)|\}$. Then F is an induced subgraph of $dS_{t,t,t}$, and hence every $(F, K_{s,s})$-free graph is also $(dS_{t,t,t}, K_{s,s})$-free.

Define $b = 3t + 1$, and let $\mathcal{S}$ consist of the vertex sets of all induced copies of $S_{t,t,t}$ in G. This is a b-system that can be constructed in polynomial time by enumerating all b-vertex subsets of $V(G)$. Apply Lemma 3.3 to $\mathcal{S}$ with parameters b, s, d. The second outcome gives d pairwise anticomplete induced copies of $S_{t,t,t}$, inducing $dS_{t,t,t}$ in G. Thus we obtain a hitting set X with $\alpha(X) \le c_{3.3}(b, s, d)$. Since X meets every induced copy of $S_{t,t,t}$, the graph $G - X$ is $S_{t,t,t}$-free.

Clearly,

$$\alpha(G, \mathsf{w}) = \max_{\substack{J \subseteq X \\ J \text{ independent}}} \left[\mathsf{w}(J) + \alpha\big(G - (X \cup N_G(J)), \mathsf{w}\big)\right].$$

The bound on $\alpha(X)$ lets us enumerate all independent subsets $J \subseteq X$ in polynomial time. For each J, the residual graph is induced in $G - X$ and hence is $(S_{t,t,t}, K_{s,s})$-free. Solve MWIS in this graph using Theorem 4.3, and add J to the resulting independent set. Returning the heaviest set obtained in this way gives a maximum-weight independent set in G. Since d, s, t are fixed, the construction of $\mathcal{S}$, the application of Lemma 3.3, and all residual computations take polynomial time. □

6 Conclusion

The main challenge remains to solve MWIS in polynomial time in graphs excluding an arbitrary fixed subdivided claw, with no further assumptions. We believe that extended strip decompositions are the right tool, but that further progress will require controlling the structure of particles, rather than just their size. The goal would be to find, in polynomial time, a decomposition whose particles admit efficient algorithms for MWIS. The structure theorem of Chudnovsky and Seymour for claw-free graphs [12] provides a model for such an approach.

Abrishami et al. [2] considered F-free graphs, for $F \in \mathcal{F}$, excluding a fixed biclique as a subgraph. By Ramsey's theorem, this restriction is equivalent, up to changing the parameters, to excluding both a fixed clique and a fixed biclique as induced subgraphs. Our result strengthens this result by removing the clique exclusion. The complementary direction is to retain only the clique exclusion. This direction was proposed by Abrishami et al. [2, Conjecture 10], and we restate it here.

Problem 6.1. *For every fixed $F \in \mathcal{F}$ and integer $s \geq 1$, can MWIS be solved in polynomial time in (F, K_s)-free graphs?*

Even the special case where F is a path remains open in general.

Problem 6.2. *For every fixed pair of integers $t, s \geq 1$, can MWIS be solved in polynomial time in (P_t, K_s)-free graphs?*

This question (in the unweighted setting) was explicitly asked by Chudnovsky, King, Pilipczuk, Rzążewski, and Spirkl [10], and remains wide open. Recall from the introduction that recently the case $t = 7$ was settled in the affirmative [9].

Acknowledgement. ChatGPT was used to simplify some arguments and to identify or correct omissions and minor mistakes, and Claude was used to polish the write-up. The authors take full responsibility for the content of the paper.

References

[1] T. Abrishami, M. Chudnovsky, C. Dibek, M. Pilipczuk, and P. Rzążewski. Max weight independent set in sparse graphs with no long claws. *ACM Trans. Algorithms*, Sept. 2026. Just Accepted.

[2] T. Abrishami, M. Chudnovsky, C. Dibek, M. Pilipczuk, and P. Rzążewski. Max weight independent set in sparse graphs with no long claws. arXiv:2309.16995v3, 2024.

[3] T. Abrishami, M. Chudnovsky, C. Dibek, and P. Rzążewski. Polynomial-time algorithm for maximum independent set in bounded-degree graphs with no long induced claws. In J. S. Naor and N. Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference, January 9–12, 2022*, pages 1448–1470. SIAM, 2022.

[4] V. E. Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. *Combinatorial-algebraic methods in applied mathematics*, pages 3–13, 1982.

[5] R. Bourneuf, J. Masaříková, W. Nadara, and M. Pilipczuk. Graphs with no long claws: An improved bound for the analog of the Gyárfás' path argument. In P. Gawrychowski, F. Mazowiecki, and M. Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, Warsaw, Poland, August 25-29, 2025*, volume 345 of *LIPIcs*, pages 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[6] A. Brandstädt, V. V. Lozin, and R. Mosca. Independent sets of maximum weight beyond claw-free graphs and related problems. *Theoretical Computer Science*, 1032:115103, 2025.

[7] A. Brandstädt and R. Mosca. Maximum weight independent sets for (P_7, triangle)-free graphs in polynomial time. *Discret. Appl. Math.*, 236:57–65, 2018.

[8] A. Brandstädt and R. Mosca. Maximum weight independent set for ℓclaw-free graphs in polynomial time. *Discrete Applied Mathematics*, 237:57–64, mar 2018.

[9] M. Chudnovsky, J. Czyżewska, K. Kluk, M. Pilipczuk, and P. Rzążewski. Sparse induced subgraphs in P_7-free graphs of bounded clique number. In *36th International Symposium on Algorithms and Computation (ISAAC 2025)*, volume 359 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.

[10] M. Chudnovsky, J. King, M. Pilipczuk, P. Rzążewski, and S. Spirkl. Finding large H-colorable subgraphs in hereditary graph classes. *SIAM Journal on Discrete Mathematics*, 35(4):2357–2386, 2021.

[11] M. Chudnovsky, M. Pilipczuk, M. Pilipczuk, and S. Thomassé. Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs. *SIAM Journal on Computing*, 53(1):47–86, 2024.

[12] M. Chudnovsky and P. D. Seymour. Claw-free graphs. V. Global structure. *J. Comb. Theory, Ser. B*, 98(6):1373–1410, 2008.

[13] Y. Faenza, G. Oriolo, and G. Stauffer. An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$-algorithm for the weighted stable set problem. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 630–646, USA, 2011. Society for Industrial and Applied Mathematics.

[14] P. Gartland and D. Lokshtanov. Independent set on P_k-free graphs in quasi-polynomial time. In S. Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16–19, 2020*, pages 613–624. IEEE, 2020.

[15] P. Gartland, D. Lokshtanov, T. Masařík, M. Pilipczuk, M. Pilipczuk, and P. Rzążewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In B. Mohar, I. Shinkar, and R. O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 683–691. ACM, 2024.

[16] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[17] A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *ACM Trans. Algorithms*, 18(1):4:1–4:57, 2022.

[18] S. Hajebi and S. Spirkl. Tree-alpha and excluding finitely many graphs. *CoRR*, abs/2605.01223, 2026.

[19] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

[20] T. Kövari, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3:50–57, 1954.

[21] D. Kühn and D. Osthus. Induced subdivisions in $K_{s,s}$-free graphs of large average degree. *Combinatorica*, 24:287–304, 2004.

[22] D. Lokshtanov, M. Vatshelle, and Y. Villanger. Independent set in P_5-free graphs in polynomial time. In C. Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014*, pages 570–581. SIAM, 2014.

[23] V. V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *J. Discrete Algorithms*, 6(4):595–604, 2008.

[24] F. Maffray and L. Pastor. Maximum weight stable set in (P_7, bull)-free graphs. *CoRR*, abs/1611.09663, 2016.

[25] K. Majewski, T. Masařík, J. Masaříková, K. Okrasa, M. Pilipczuk, P. Rzążewski, and M. Sokołowski. Max weight independent set in graphs with no long claws: An analog of the Gyárfás' path argument. *ACM Transactions on Computation Theory*, 16(2):8:1–8:18, Mar. 2024.

[26] G. J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980.

[27] R. Mosca. Stable sets of maximum weight in (P_7, banner)-free graphs. *Discrete Mathematics*, 308(1):20–33, 2008.

[28] M. Pilipczuk, M. Pilipczuk, and P. Rzążewski. Quasi-polynomial-time algorithm for independent set in P_t-free graphs via shrinking the space of induced paths. In H. V. Le and V. King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11–12, 2021*, pages 204–209. SIAM, 2021.

[29] N. Sbihi. Algorithme de recherche d'un stable de cardinalite maximum dans un graphe sans etoile. *Discrete Mathematics*, 29(1):53–76, 1980.

[30] G. Valiente. A new simple algorithm for the maximum-weight independent set problem on circle graphs. In *Algorithms and Computation, ISAAC 2003*, volume 2906 of *Lecture Notes in Computer Science*, pages 129–137. Springer, 2003.