Max Weight Independent Set in sparse graphs with no long claws*

Tara Abrishami[†] Maria Chudnovsky[‡] Cemil Dibek[§] Marcin Pilipczuk[¶]
Paweł Rzążewski[‖]

Abstract

For graphs G and H, we say that G is H-free if it does not contain H as an induced subgraph. Already in the early 1980s Alekseev observed that the Max Weight Independent Set problem (MWIS) remains NP-hard in H-free graphs, unless every component of H is a path or a subdivided claw, i.e., a graph obtained from the three-leaf star by subdividing each edge some number of times (possibly zero). Since then, determining the complexity of MWIS in these remaining cases is one of the most important problems in algorithmic graph theory.

In this paper we make an important step towards solving the problem by providing a polynomial-time algorithm for MWIS in graphs excluding a fixed graph forest of paths and subdivided claws as an induced subgraph, and a fixed biclique as a subgraph.

1 Introduction

A *vertex-weighted graph* is an undirected graph G equipped with a weight function $\mathbf{w} : V(G) \to \mathbb{N}$. For $X \subseteq V(G)$, we use $\mathbf{w}(X)$ as a shorthand for $\sum_{x \in X} \mathbf{w}(x)$ and for a subgraph H of G, $\mathbf{w}(H)$ is a shorthand for $\mathbf{w}(V(H))$. By convention we use $\mathbf{w}(\emptyset) = 0$. Throughout the paper, we assume that arithmetic operations on weights are performed in unit time.

For a graph G, a set $I \subseteq V(G)$ is *independent* or *stable* if there is no edge of G with both endpoints in I. By $\alpha(G)$ we denote the number of vertices in a largest independent set in G. In the Max Independent Set (MIS) problem, we are given an undirected graph G, and ask for an independent set of size $\alpha(G)$. In the Max Weight Independent Set (MWIS) problem we are given an undirected vertex-weighted graph $(G, \mathbf{w})$ and ask for a maximum-weight independent set in $(G, \mathbf{w})$. Note that MIS is a special case of MWIS where all weights are equal. By n we always denote the number of vertices of the instance graph.

MIS (and MWIS as its generalization) is a "canonical" hard problem: It was one of the first problems shown to be NP-hard [30], it is notoriously hard to approximate [29, 32], and it is W[1]-hard [19]. Many of these hardness results hold even if we restrict input instances to some natural graph classes [9, 20, 22]. Thus, a very natural research direction is to consider restricted instances and try to capture the boundary between "easy" and "hard" cases.

---

*Parts of this work appeared in the proceedings of SODA 2022 [1] and STACS 2024 [3]

[†]Department of Mathematics, University of Hamburg, Germany, tara.abrishami@uni-hamburg.de. Supported by the National Science Foundation Award Number DMS-2303251 and the Alexander von Humboldt Foundation. This work was performed in part while the author was at Princeton University and supported by NSF Grant DMS-1763817 and NSF-EPSRC Grant DMS-2120644.

[‡]Department of Mathematics, Princeton University, USA, mchudnov@math.princeton.edu. Supported by NSF-EPSRC Grant DMS-2120644 and by AFOSR grant FA9550-22-1-008.

[§]Department of Industrial Engineering, Koç University, Turkey, cdibek@ku.edu.tr. Supported by the National Science Foundation Award Number DMS-1763817. This work was performed in part while the author was at Princeton University.

[¶]Institute of Informatics, University of Warsaw, Poland, m.pilipczuk@mimuw.edu.pl. Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.

[‖]Faculty of Mathematics and Information Science, Warsaw University of Technology and Institute of Informatics, University of Warsaw, Poland, pawel.rzazewski@pw.edu.pl. This work is a part of the project BOBR that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 948057)

State of the art. The study of the complexity of MWIS in restricted graph classes is a central topic in algorithmic graph theory [2, 5, 8, 17, 26, 54]. Particular attention is given to classes that are *hereditary*, i.e., closed under vertex deletion. Among such classes a special role is played by ones defined by forbidding certain substructures. For graphs G and H, we say that G is H*-free* if it does not contain H as an *induced subgraph*.

In what follows, for $t \geqslant 1$, by P_t we denote the t-vertex path. For integers $a, b, c \geqslant 1$, by $S_{a,b,c}$ we denote the graph obtained from the three-leaf star (i.e., the *claw*) by subdividing the three edges $a-1$, $b-1$, and $c-1$ times, respectively. Alternatively, we may think of $S_{a,b,c}$ as the graph obtained from paths P_{a+1}, P_{b+1}, and P_{c+1} by identifying one endvertex of each path. By $dS_{a,b,c}$ we denote the graph with d components, each of which is isomorphic to $S_{a,b,c}$.

Let $\mathcal{S}$ be the family of subcubic forests H whose every component has at most one vertex of degree 3, i.e., whose every component is either a path or a subdivided claw.

The complexity study of MWIS in H-free graphs dates back to the early 1980s and the work of Alekseev [4], who observed that for most graphs H the problem remains NP-hard. Indeed, let us discuss the hard cases. First, MIS (and thus MWIS) is NP-hard in subcubic graphs [22], which are H-free whenever H has a vertex of degree at least 4. For the remaining cases we will use the so-called *Poljak construction* [51]: If G' is obtained from G by subdividing one edge twice, then $\alpha(G') = \alpha(G) + 1$. Thus, if G^p denotes the graph obtained from G by subdividing each edge exactly $2p$ times, then $\alpha(G^p) = \alpha(G) + p \cdot |E(G)|$. Now observe that if H has a cycle or two vertices of degree three in one component, then $G^{|V(H)|}$ is H-free. Consequently, for such graph H, MIS is NP-hard in H-free graphs. Let us point out that the above hardness reductions imply that MIS cannot even be solved in subexponential time unless the Exponential-Time Hypothesis (ETH) fails.

Summing up, the only graphs H for which we may hope for a polynomial-time (or even subexponential-time) algorithm for MWIS in H-free graphs are precisely the graphs in $\mathcal{S}$.

The complexity of MWIS in H-free graphs when $H \in \mathcal{S}$ remains one of the most challenging and important problems in algorithmic graph theory. Despite significant attention received from the graph theory and the theoretical computer science communities, only partial results are known. Let us discuss them.

First, consider the case that $H = P_t$ for some t. Since P_4-free graphs, also known as *cographs*, have very rigid structure (in particular, they have clique-width at most 2), the polynomial-time algorithm for this class of graphs is rather simple [18]. However, already for P_5-free graphs the situation is much more complicated. The existence of a polynomial-time algorithm for P_5-free graphs was a long-standing open problem that was finally resolved in the affirmative in 2014 by Lokshtanov, Vatshelle, and Villanger [35] using the framework of *potential maximal cliques*. Later, using the same approach but with significantly more technical effort, Grzesik, Klimošová, Pilipczuk, and Pilipczuk [27] obtained a polynomial-time algorithm for P_6-free graphs. The case of P_7-free graphs remains open.

However, some interesting algorithmic results can be obtained if we relax our notion of an efficient algorithm. First, it was shown by Bacsó et al. [7] that for every fixed t, MWIS can be solved in *subexponential* time $2^{\mathcal{O}(\sqrt{n \log n})}$ for P_t-free graphs. Another subexponential-time algorithm, with worse running time, was obtained independently by Brause [13]. While these results do not rule out the possibility that the problem is NP-hard, let us recall that, assuming the ETH, subexponential algorithms for MWIS in H-free graphs cannot exist if $H \notin \mathcal{S}$. Later, Chudnovsky et al. [15] showed that for every fixed t, the problem admits a QPTAS in P_t-free graphs. Finally, a very recent breakthrough result by Gartland and Lokshtanov [23] shows that for every fixed t, the problem can be solved in *quasipolynomial time* $n^{\mathcal{O}(\log^3 n)}$; see also a slightly simpler algorithm by Pilipczuk, Pilipczuk, and Rzążewski [50] with running time $n^{\mathcal{O}(\log^2 n)}$. Note that this means that if for some t, MWIS is NP-hard for P_t-free graphs, then *all problems* in NP can be solved in quasipolynomial time. While this does yet not imply that P = NP, it still seems rather unlikely according to our current understanding of complexity theory.

Now let us turn to the case when H is a subdivided claw. The simplest subdivided claw is the claw itself, i.e., $S_{1,1,1} = K_{1,3}$. Claw-free graphs appear to be closely related to *line graphs* [16] and thus a polynomial-time algorithm for MWIS in claw-free graphs can be obtained by a modification of the well-known augmenting path approach for finding a maximum-weight matching [43, 49, 52] (i.e., a maximum-weight independent set in a line graph). Let us highlight the close relation of claw-free graphs and line graphs, as it will play an important role in our paper. The next smallest subdivided claw is the *fork*, i.e., $S_{2,1,1}$. A polynomial-time algorithm for MIS in fork-free graphs was obtained by Alekseev [6].

Later it was extended to the MWIS problem by Lozin and Milanič [36]. For disconnected H, it is known that MWIS is polynomial-time-solvable in $dS_{1,1,1}$-free graphs, for every fixed d [10]. The existence of polynomial-time algorithms in the next simplest (connected) cases, i.e., $H = S_{3,1,1}$ and $H = S_{2,2,1}$, is wide open.

Again, some interesting results can be obtained if we look beyond polynomial-time algorithms. Chudnovsky et al. [15] proved that for every subdivided claw H, the MWIS problem in H-free graphs admits a QPTAS and a subexponential-time algorithm working in time $n^{\mathcal{O}(n^{8/9})}$. We point out that the arguments used for the case when H is a subdivided claw are significantly more complicated and technically involved than their counterparts for P_t-free graphs. These results were then simplified and improved by Majewski et al. [42]: They obtained another (faster) QPTAS and a subexponential-time algorithm with running time $n^{\mathcal{O}(\sqrt{n}\log n)}$. The pinnacle of this line of research is the quasipolynomial-time algorithm for MWIS in H-free graphs, for every $H \in \mathcal{S}$, given by Gartland et al. [24].

More tractability results can be obtained if we put some additional restrictions on the instance graph. In particular, there is a long line of research concerning graphs excluding a fixed (but still small) path or a subdivided claw and, simultaneously, some other small graphs, see e.g. [11, 12, 25, 28, 34, 37–39, 41, 44–48]. A slightly different direction was considered by Lozin, Milanič, and Purcell [37], who proved that for every fixed t, MWIS is polynomial-time solvable in *subcubic* $S_{t,t,1}$-free graphs. Later, Lozin, Monnot, and Ries [38] showed a polynomial time algorithm for subcubic $S_{2,2,2}$-free graphs. Finally, Harutyunyan et al. [28] generalized both these results by providing a polynomial-time algorithm for subcubic $S_{t,t,2}$-free graphs, for any fixed t.

We remark that the case when H is a subdivided claw (or, more precisely, is in $\mathcal{S}$ and contains at least one component which is not a path) is the only case where the restriction to bounded degree graphs leads to an interesting problem. Indeed, the already mentioned hardness reduction of Alekseev [4] shows that if $H \notin \mathcal{S}$, then MIS is NP-hard even in *subcubic* H-free graphs. On the other hand, if H is a forest of paths, then connected H-free graphs of bounded degree are of constant size and thus of little interest.

In this work, we continue and significantly extend the study of the complexity of MWIS in H-free graphs with additional restrictions, where $H \in \mathcal{S}$.

Our results. As a warm-up, we present a polynomial-time algorithm for H-free graphs of bounded degree, where $H \in \mathcal{S}$.

Theorem 1 *There exists an algorithm that, given a vertex-weighted graph $(G, \mathbf{w})$ on n vertices with maximum degree Δ and integers d, t in time $2^{\mathcal{O}(dt\Delta^2)} n^{\mathcal{O}(t\Delta^2)}$ either finds an induced $dS_{t,t,t}$ or the maximum possible weight of an independent set in $(G, \mathbf{w})$.*

Note that by picking appropriate d and t, Theorem 1 yields a polynomial-time algorithm for MWIS for bounded-degree graphs excluding a fixed graph from $\mathcal{S}$ as an induced subgraph.

Then we proceed to the main result of the paper: we show that MWIS remains polynomial-time solvable in $dS_{t,t,t}$-free graphs, even if instead of bounding the maximum degree, we forbid a fixed biclique *as a subgraph.*

Theorem 2 *For every fixed integers d, t, and s there exists a polynomial-time algorithm that, given a vertex-weighted graph $(G, \mathbf{w})$ that does not contain $dS_{t,t,t}$ as an induced subgraph nor $K_{s,s}$ as a subgraph, returns the maximum possible weight of an independent set in $(G, \mathbf{w})$.*

Let us remark that by the celebrated Kövári-Sós-Turán theorem [33], classes that exclude $K_{s,s}$ as a subgraph capture all hereditary classes of *sparse graphs*, where by "sparse" we mean that the graph has a subquadratic number of edges. Furthermore, by a simple Ramsey argument, for every positive integer r there exists an integer s such that if G is K_r-free and $K_{r,r}$-free, then G does not contain $K_{s,s}$ as a subgraph. Hence, equivalently, Theorem 2 yields a polynomial-time algorithm for MWIS for graphs that are simultaneously H-free (for some $H \in \mathcal{S}$), K_r-free, and $K_{r,r}$-free.

Our techniques. As in the previous works [15, 24], the crucial tool in handling $dS_{t,t,t}$-free graphs is an *extended strip decomposition.* Its technical definition can be found in preliminaries; for now, it suffices to say that it is a wide generalization of the preimage graph of a line graph (recall that line graphs are $S_{1,1,1}$-free) that allows for recursion for the MWIS problem. An extended strip decomposition of a graph

G identifies some induced subgraphs of G as *particles* and, knowing the maximum possible weight of an independent set in each particle, one can compute in polynomial time the maximum possible weight of an independent set in G. (We remark that this computation involves advanced combinatorial techniques as it relies on a reduction to the maximum weight matching problem in an auxiliary graph.) In other words, finding an extended strip decomposition with small particles compared to $|V(G)|$ is equally good for the MWIS problem as splitting the graph into small connected components.

The starting point is the following theorem of [42].

Theorem 3 ([42, Corollary 12] in a semi-weighted setting) *There exists an algorithm that, given an n-vertex graph G with a set $U \subseteq V(G)$ and integers d, t, in polynomial time outputs either:*

- *an induced copy of $dS_{t,t,t}$ in G, or*
- *a set X of size at most $(d-1)(3t+1)+(11\log n+6)(t+1)$ and a rigid extended strip decomposition of $G - N[X]$ with every particle containing at most $|U|/2$ vertices of U.*

(A rigid extended strip decomposition is an extended strip decomposition that does not have some unnecessary empty sets. By $N[X]$ we denote the set consisting of X and all vertices with a neighbor in X.) Let us remark that the result stated in [42, Theorem 2] is for unweighted graphs (i.e., $U = V(G)$ using the notation from Theorem 3), but the statement of Theorem 3 can be easily derived from the proof, see also [24].

Consider the setting of Theorem 1, i.e., the graph G has maximum degree Δ. We apply Theorem 3 to G with $U = V(G)$. If we get the first outcome, i.e., an induced $dS_{t,t,t}$ in G, we return it and terminate. So assume that we get the second outcome, i.e., the set X. Note that as $|X| = \mathcal{O}(dt + t\log n)$, we have $|N[X]| = \mathcal{O}(dt\Delta + t\Delta\log n)$. It is now tempting to exhaustively branch on $N[X]$ (i.e., guess the intersection of the sought independent set with $N[X]$) and recurse on the particles of the extended strip decomposition of $G - N[X]$. However, implementing this strategy directly gives quasipolynomial (in n) running time bound of $n^{\mathcal{O}(dt\Delta + t\Delta\log n)}$, as the branching step yields up to $2^{|N[X]|} = 2^{\mathcal{O}(dt\Delta)} \cdot n^{\mathcal{O}(t\Delta)}$ subcases and the depth of the recursion is $\mathcal{O}(\log n)$.

Our main new idea now is to perform this branching lazily, by considering a more general *border* version of the problem, where the input graph is additionally equipped with a set of *terminals* and we ask for a maximum weight of an independent set for every possible behavior on the terminals.

Border MWIS

Input:	A vertex-weighted graph $(G, \mathbf{w})$ with a set $T \subseteq V(G)$ of *terminals*.
Task:	Compute $f_{G,\mathbf{w},T} : 2^T \to \mathbb{N} \cup \{-\infty\}$ defined for every $I_T \subseteq T$ as $f_{G,\mathbf{w},T}(I_T) = \max\{\mathbf{w}(I) \mid I \subseteq V(G) \wedge I \text{ is independent} \wedge I \cap T = I_T\}$.

A similar application of a border version of the problem to postpone branching in recursion appeared for example in the technique of recursive understanding [14, 31].

Let us return to our setting, where we have a set X of size $\mathcal{O}(dt + t\log n)$ and an extended strip decomposition of $G - N[X]$ with particles of size at most half of the size of $V(G)$. We would like to remove $N[X]$ from the graph, indicate $N(N[X])$ as terminals and solve Border MWIS in $(G - N[X], \mathbf{w}, T := N(N[X]))$ using the extended strip decomposition for recursion. Note that, thanks to the bounded degree assumption, the size of $T = N(N[X])$ is bounded by $\mathcal{O}(dt\Delta^2 + t\Delta^2\log n)$.

This approach *almost* works: the only problem is that, as the recursion progresses, the set of terminals accummulates and its size can grow beyond the initial $\mathcal{O}(dt\Delta^2 + t\Delta^2\log n)$ bound. Luckily, this can be remedied in a standard way: we alternate recursive steps where Theorem 3 is invoked with $U = V(G)$ with steps where Theorem 3 is invoked with $U = T$. In this manner, we can maintain a bound of $\mathcal{O}(dt\Delta^2 + t\Delta^2\log n)$ on the number of terminals in every recursive call. Note that this bound also guarantees that the size of the domain of the requested function $f_{G,\mathbf{w},T}$ is of size $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$, which is within the promised time bound.

Let us now move to the more general setting of Theorem 2. Here, the starting points are the recent results of Weißauer [53] and Lozin and Razgon [40] that show that in the $S_{t,t,t}$-free case, excluding a biclique as a subgraph is not that much different than bounding the maximum degree.

A *k-block* in a graph G is a set B of k vertices, no two of which can be separated by deleting fewer than k vertices. More precisely, there is no set X of size less than k, such that more than one component of $G - X$ contains a vertex from B. The following result was shown by Weißauer (we refer to preliminaries for standard definitions of tree decompositions, adhesions, and torsos).

Theorem 4 (Weißauer [53]) *Let G be a graph and $k \geqslant 2$ be an integer. If G has no $(k+1)$-block, then G admits a tree decomposition with every adhesion of size at most k, in which every torso has at most k vertices of degree at least $2k(k-1)$.*

Even though the statement of the result in [53] is just existential, the proof actually yields a polynomial-time algorithm to compute such a tree decomposition.

It turns out that $dS_{t,t,t}$-free graphs with no large bicliques have no large blocks.

Lemma 5 *For any d, t, and s there exists k such that the following holds. Every $dS_{t,t,t}$-free graph with no subgraph isomorphic to $K_{s,s}$ has no k-block.*

Let us remark that Lozin and Razgon [40] showed Lemma 5 for $S_{t,t,t}$-free graphs. However, an extension of their argument applies to $dS_{t,t,t}$-free graphs; we include it in Appendix A.

Combining Theorem 4 and Lemma 5 we immediately obtain the following.

Corollary 6 *For any d, t, and s there exists k such that the following holds. Given a $dS_{t,t,t}$-free graph G with no subgraph isomorphic to $K_{s,s}$, in polynomial time one can compute a tree decomposition of G with each adhesion of size at most k, in which every torso has at most k vertices of degree at least $2k(k-1)$.*

To prove Theorem 2 using Corollary 6 we need to carefully combine explicit branching on the (bounded number of) vertices of large degree in a single bag with — as in the bounded degree case — applying Theorem 3 to the remainder of the graph and indicating $N(N[X])$ as the terminal set of the border problem passed to the recursive calls. Finally, we combine these steps with the information passed between the bags of the tree decomposition.

2 Preliminaries

Our algorithms take a vertex-weighted graph $(G, \mathbf{w})$ as an input. In the recursion, we will be working on various induced subgraphs of G with vertex weight inherited from $\mathbf{w}$. Somewhat abusing notation, we will keep $\mathbf{w}$ for the weight function in any induced subgraph of G.

Tree decompositions. Let G be a graph. A *tree decomposition* of G is a pair $(\mathcal{T}, \beta)$ where $\mathcal{T}$ is a tree and $\beta : V(\mathcal{T}) \to 2^{V(G)}$ is a function satisfying the following: (i) for every $uv \in E(G)$ there exists $t \in V(\mathcal{T})$ with $u, v \in \beta(t)$, and (ii) for every $v \in V(G)$ the set $\{t \in V(\mathcal{T}) \mid v \in \beta(t)\}$ induces a connected nonempty subtree of $\mathcal{T}$. For every $t \in V(\mathcal{T})$ and $st \in E(\mathcal{T})$, the set $\beta(t)$ is the *bag* at node t and the set $\sigma(st) := \beta(s) \cap \beta(t)$ is the *adhesion at edge st*. An *adhesion* of a tree decomposition $(\mathcal{T}, \beta)$ is the adhesion at some edge of $\mathcal{T}$. The critical property of a tree decomposition $(\mathcal{T}, \beta)$ is that if $st \in E(\mathcal{T})$ and V_s and V_t are two connected components of $\mathcal{T} - \{st\}$ that contain s and t, respectively, then $\sigma(st)$ separates $\bigcup_{x \in V_s} \beta(x) \setminus \sigma(st)$ from $\bigcup_{x \in V_t} \beta(x) \setminus \sigma(st)$ in G.

The *torso* of a bag $\beta(t)$ in a tree decomposition $(\mathcal{T}, \beta)$ is a graph H with $V(H) = \beta(t)$ and $uv \in E(H)$ if $uv \in E(G)$ or there exists a neighbor $s \in N_{\mathcal{T}}(t)$ with $u, v \in \sigma(st)$. That is, the torso of $\beta(t)$ is created from $G[\beta(t)]$ by turning the adhesion $\sigma(st)$ into a clique for every neighbor s of t in $\mathcal{T}$.

Extended strip decompositions. We follow the notation of [24, 42]. 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. Similarly as we write xy instead of $\{x, y\}$ for an edge, we write xyz instead of $\{x, y, z\}$ for a triangle. An *extended strip decomposition* of a graph G is a pair (H, η) that consists of:

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

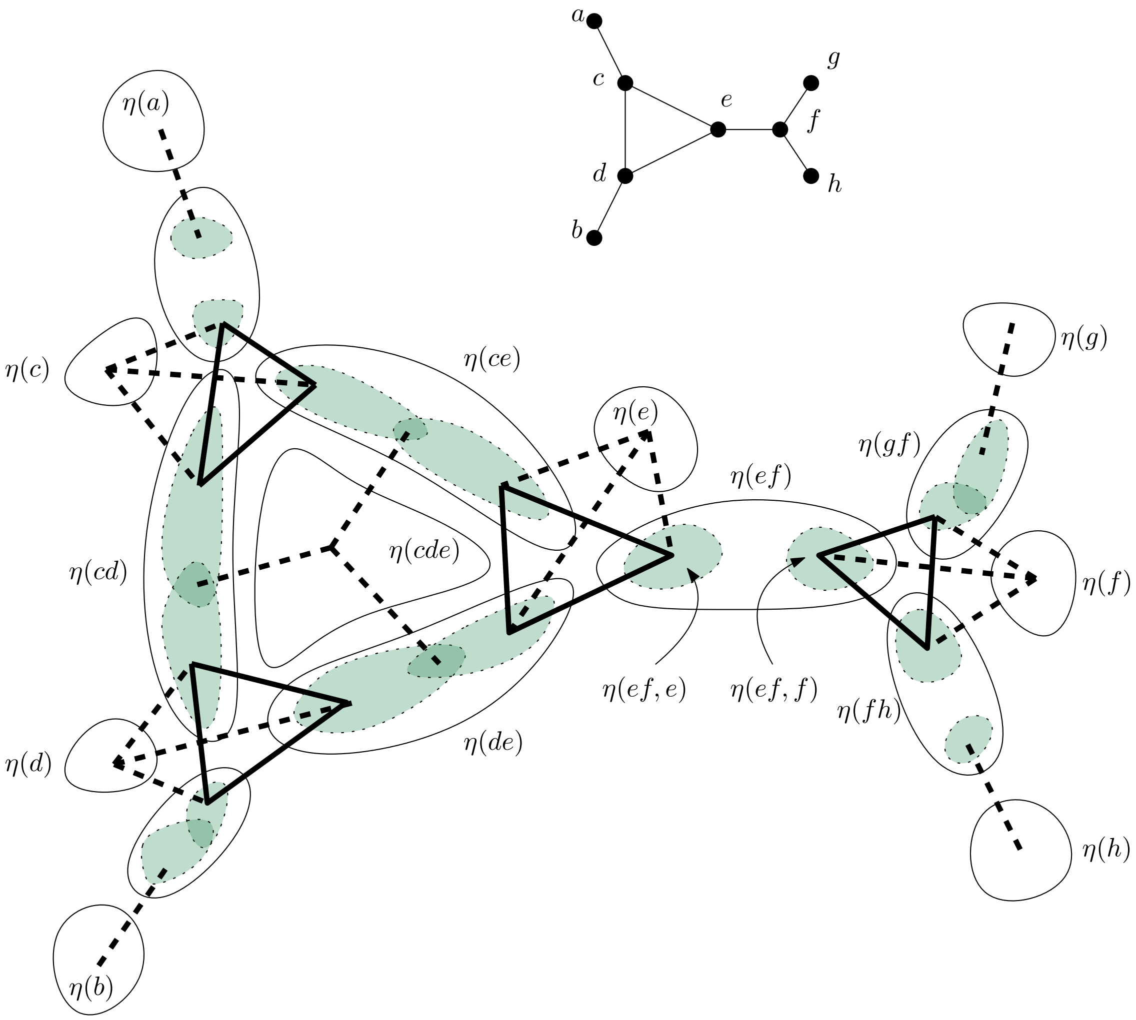

Figure 1: A graph H (top) and a schematic view of an extended strip decomposition (H, η) of G. Green areas depict the sets $\eta(xy, y)$. Solid line indicates that all edges between two sets exist. Dashed line denotes that edges might exist. Picture from [24], courtesy of the authors.

which satisfy the following properties:

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 sets $\eta(o)$ 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 **??**.

An extended strip decomposition (H, η) is *rigid* if for every $xy \in E(H)$, the sets $\eta(xy)$, $\eta(xy, x)$, and $\eta(xy, y)$ are nonempty, and for every isolated $x \in V(H)$, the set $\eta(x)$ is nonempty. Note that if (H, η) is a rigid extended strip decomposition of G, then $|V(H)|$ is bounded by $|V(G)|$.

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

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

As announced in the introduction, to solve MWIS in G it suffices to know the solution to MWIS in particles. The proof of the following lemma follows closely the lines of the proof of analogous statement of [15] and is included for completeness in Appendix B.

Lemma 7 *Given a* Border MWIS *instance* $(G, \mathbf{w}, T)$*, an extended strip decomposition* (H, η) *of* G*, and a solution* $f_{G[A],\mathbf{w},T\cap A}$ *to the* Border MWIS *instance* $(G[A], \mathbf{w}, T \cap A)$ *for every particle* A *of* (H, η)*, one can in time* $2^{|T|}$ *times a polynomial in* $|V(G)| + |V(H)|$ *compute the solution* $f_{G,\mathbf{w},T}$ *to the input* $(G, \mathbf{w}, T)$*.*

We need the following simple observations.

Lemma 8 *Let* G *be a* K_t*-free graph and let* (H, η) *be a rigid extended strip decomposition of* G*. Then the maximum degree of* H *is at most* $t - 1$*.*

Proof. Let $x \in V(H)$. Observe that the sets $\{\eta(xy, x) \mid y \in N_H(x)\}$ are nonempty and complete to each other in G. Hence, G contains a clique of size equal to the degree of x in H. □

Lemma 9 *Let* G *be a graph and let* (H, η) *be an extended strip decomposition of* G *such that the maximum degree of* H *is at most* d*. Then, every vertex of* G *is in at most* $\max(4, 2d + 1)$ *particles.*

Proof. Pick $v \in V(G)$ and observe that:

- If $v \in \eta(x)$ for some $x \in V(H)$, then v is in the vertex particle of x and in one half-edge and one full-edge particle for every edge of H incident with x. Since there are at most d such edges, v is in at most $2d + 1$ particles.
- If $v \in \eta(xy)$ for some $xy \in E(H)$, then v is in at most four particles for the edge xy.
- If $v \in \eta(xyz)$ for some $xyz \in T(H)$, then v is in the triangle particle for xyz and in three full edge particles, for the three sides of the triangle xyz. □

3 Bounded-degree graphs: Proof of Theorem 1

This section is devoted to the proof of Theorem 1.

Let d, t be positive integers and let $(G, \mathbf{w})$ be the input vertex-weighted graph. We denote $n := |V(G)|$ and Δ to be the maximum degree of G. Let

$$\ell := (d - 1)(3t + 1) + \lceil 11 \log n + 6\rceil(t + 2) = \mathcal{O}(dt + t \log n)$$

be an upper bound on the size of X for any application of Theorem 3 for any induced subgraph of G.

We describe a recursive algorithm that takes as input an induced subgraph G' of G with weights $\mathbf{w}$ and a set of terminals $T \subseteq V(G')$ of size at most $4\ell\Delta^2$ and solves Border MWIS on $(G', \mathbf{w}, T)$. The root call is for $G' := G$ and $T := \emptyset$; indeed, note that $f_{G,\mathbf{w},\emptyset}(\emptyset)$ is the maximum possible weight of an independent set in G.

Let $(G', \mathbf{w}, T)$ be an input to a recursive call. First, the algorithm initializes $f_{G',\mathbf{w},T}(I_T) := -\infty$ for every $I_T \subseteq T$.

If $|V(G')| \leqslant 4\Delta^2\ell$, the algorithm proceeds by brute-force: it enumerates independent sets $I \subseteq V(G')$ and updates $f_{G',\mathbf{w},T}(I \cap T)$ with $\mathbf{w}(I)$ whenever the previous value of that cell was smaller. As $\ell =$

$\mathcal{O}(dt + t\log n)$, this step takes $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$ time. This completes the description of the leaf step of the recursion.

If $|V(G')| > 4\Delta^2\ell$, the algorithm proceeds as follows. If $|T| \leqslant 3\Delta^2\ell$, let $U := V(G')$, and otherwise, let $U := T$. The algorithm invokes Theorem 3 on G' and U. If an induced $dS_{t,t,t}$ is returned, then it can be returned by the main algorithm as it is in particular an induced subgraph of G. Hence, we can assume that we obtain a set $X \subseteq V(G)$ of size at most ℓ and an extended strip decomposition (H, η) of $G^* := G' - N_{G'}[X]$ whose every particle contains at most $|U|/2$ vertices of U.

Observe that as $|X| \leqslant \ell$ and the maximum degree of G is Δ, we have $|N_{G'}(N_{G'}[X])| \leqslant \Delta^2\ell$. Let $T^* := (T \cap V(G^*)) \cup N_{G'}(N_{G'}[X])$. Note that we have $T^* \subseteq V(G^*)$ and $|T^*| \leqslant 5\Delta^2\ell$. For every particle A of (H, η), we invoke a recursive call on $(G^*_A := G^*[A], \mathbf{w}, T^*_A := T^* \cap A)$, obtaining $f_{G^*_A,\mathbf{w},T^*_A}$ (or an induced $dS_{t,t,t}$, which can be directly returned). We use Lemma 7 to obtain a solution $f_{G^*,\mathbf{w},T^*}$ to Border MWIS instance $(G^*, \mathbf{w}, T^*)$.

Finally, we iterate over every $I_T \subseteq T^* \cup N_{G'}[X]$ (note that $T \subseteq T^* \cup N_{G'}[X]$) and, if I_T is independent, update the cell $f_{G',\mathbf{w},T}(I_T \cap T)$ with the value $\mathbf{w}(I_T \setminus T^*) + f_{G^*,\mathbf{w},T^*}(I_T \cap T^*)$, if this value is larger than the previous value of this cell. This completes the description of the algorithm.

The correctness of the algorithm is immediate thanks to Lemma 7 and the fact that $N_{G'}[X]$ is adjacent in G' only to $N_{G'}(N_{G'}[X])$, which is a subset of T^*.

For the complexity analysis, consider a recursive call to $(G^*_A, \mathbf{w}, T^*_A)$ for a particle A. If $|T| \leqslant 3\Delta^2\ell$, then $|T^*_A| \leqslant |T^*| \leqslant 4\Delta^2\ell$. Otherwise, $U = T$ and $|T \cap A| \leqslant |T|/2 \leqslant 2\Delta^2\ell$. As $|N_{G'}(N_{G'}[X])| \leqslant \Delta^2\ell$, we have $|T^*_A| \leqslant 3\Delta^2\ell$. Hence, in the recursive call the invariant of at most $4\Delta^2\ell$ terminals is maintained and, moreover:

- if $|T| \leqslant 3\Delta^2\ell$, then $U = V(G')$ and $|V(G^*_A)| = |A| \leqslant |V(G')|/2$;
- otherwise, $V(G^*_A) \subseteq V(G')$ and $|T^*_A| \leqslant 3\Delta^2\ell$, hence the recursive call will fall under the first bullet.

We infer that the depth of the recursion is at most $2\lceil \log n \rceil$.

At every non-leaf recursive call, we spend $n^{\mathcal{O}(1)}$ time on invoking the algorithm from Theorem 3, $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$ time to compute $f_{G^*,\mathbf{w},T^*}$ using Lemma 7, and $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$ time for the final iteration over all subsets $I_T \subseteq T^* \cup N_{G'}[X]$. Hence, the time spent at every recursive call is bounded by $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$.

At every non-leaf recursive call, we make subcalls to $(G^*_A, \mathbf{w}, T^*_A)$ for every particle A of (H, η). Lemmas 8 and 9 ensure that the sum of $|V(G^*_A)|$ over all particles A is bounded by $(2\Delta + 3)|V(G')|$. Hence, the total size of all graphs in the i-th level of the recursion is bounded by $n \cdot (2\Delta + 3)^i$. Since the depth of the recursion is bounded by $2\lceil \log n \rceil$, the total size of all graphs in the recursion tree is bounded by $n^{\mathcal{O}(\log \Delta)}$. Since this also bounds the size of the recursion tree, we infer that the whole algorithm runs in time $2^{\mathcal{O}(dt\Delta^2)}n^{\mathcal{O}(t\Delta^2)}$.

This completes the proof of Theorem 1.

4 Graphs with no large bicliques: Proof of Theorem 2

This section is devoted to the proof of Theorem 2.

Let d, t, s be positive integers and let k be the constant depending on d, t, s from Corollary 6. Note that we can assume that $k \geqslant 2$. Again, let $(G, \mathbf{w})$ be the input vertex-weighted graph, let $n := |V(G)|$, and let

$$\ell := (d-1)(3t+1) + \lceil 11 \log n + 6 \rceil (t+2) = \mathcal{O}(dt \log n)$$

be an upper bound on the size of X for any application of Theorem 3 for any induced subgraph of G.[1]

The general framework and the leaves of the recursion are almost exactly the same as in the previous section, but with different thresholds. That is, we describe a recursive algorithm that takes as input an induced subgraph G' of G with weights $\mathbf{w}$ and a set of terminals $T \subseteq V(G')$ of size at most $32k^6\ell$ and solves Border MWIS on $(G', \mathbf{w}, T)$. The root call is for $G' := G$ and $T := \emptyset$ and the algorithm returns $f_{G,\mathbf{w},\emptyset}(\emptyset)$ as the final answer.

Let $(G', \mathbf{w}, T)$ be an input to a recursive call. The algorithm initiates first $f_{G',\mathbf{w},T}(I_T) = -\infty$ for every $I_T \subseteq T$.

[1] As the dependence of k on d, t, s is superpolynomial, for the sake of simplicity, we do not try to optimize the dependence of the complexity bound on d, t, s.

If $|V(G')| \leqslant 32k^6\ell$, the algorithm proceeds by brute-force: it enumerates independent sets $I \subseteq V(G')$ and updates $f_{G',\mathbf{w},T}(I \cap T)$ with $\mathbf{w}(I)$ whenever the previous value of that cell was smaller. As $\ell = \mathcal{O}(dt \log n)$ and k is a constant depending on d, t, and s, this step takes polynomial time. This completes the description of the leaf step of the recursion.

Otherwise, if $|V(G')| > 32k^6\ell$, we invoke Corollary 6 on G', obtaining a tree decomposition $(\mathcal{T}, \beta)$ of G'. If $|T| \leqslant 24k^6\ell$, let $U := V(G') \setminus T$, and otherwise, let $U := T$.

For every $t_1t_2 \in E(\mathcal{T})$, we proceed as follows. For $i = 1, 2$, let $\mathcal{T}_i$ be the connected component of $\mathcal{T} - \{t_1t_2\}$ that contains t_i and let $V_i = \bigcup_{x \in \mathcal{T}_i} \beta(x) \setminus \sigma(t_1t_2)$. Clearly, $\sigma(t_1t_2)$ separates V_1 from V_2. We orient the edge t_1t_2 towards t_i with larger $|U \cap V_i|$, breaking ties arbitrarily.

There exists $t \in V(\mathcal{T})$ of outdegree 0. Then, for every connected component C of $G' - \beta(t)$ we have $|C \cap U| \leqslant |U|/2$. Fix one such node t and let $B := \beta(t)$ and let $\mathcal{C}$ be the set of connected components of $G' - B$. Let G^B be a supergraph of $G'[B]$ obtained from $G'[B]$ by turning, for every $C \in \mathcal{C}$, the neighborhood $N_{G'}(C)$ into a clique. Note that G^B is a subgraph of the torso of $\beta(t)$. Hence, by the properties promised by Corollary 6, for every $C \in \mathcal{C}$ we have $|N_{G'}(C)| \leqslant k$ (as this set is contained in a single adhesion of an edge incident with t in $\mathcal{T}$) and G^B contains at most k vertices of degree at least $2k(k-1)$. Let Q be the set of vertices of G^B of degree at least $2k(k-1)$.

We perform exhaustive branching on Q. That is, we iterate over all independent sets $J \subseteq Q$ and denote $G^J := G' - Q - N_{G'}(J)$, $T^J := T \cap V(G^J)$, $U^J := U \cap V(G^J)$. For one J, we proceed as follows.

We invoke Theorem 3 to G^J with set U^J, obtaining a set X^J of size at most ℓ and a rigid extended strip decomposition (H^J, η^J) of $G^J - N_{G^J}[X^J]$ whose every particle has at most $|U^J|/2 \leqslant |U|/2$ vertices of U. Note that G^J is an induced subgraph of G', which is an induced subgraph of G, so there is no induced $dS_{t,t,t}$ in G^J.

A component $C \in \mathcal{C}$ is *dirty* if $N_{G^J}[X^J] \cap N_{G'}[C] \neq \emptyset$ and *clean* otherwise. Let

$$Y^J := (N_{G^J}[X^J] \cap B) \cup \bigcup_{C \in \mathcal{C}\colon C \text{ is dirty}} (N_{G'}(C) \cap V(G^J)).$$

The following bounds will be important for further steps.

$$|N_{G^J}[X^J] \cap B| \leqslant 2k(k-1)|X^J|. \tag{1}$$

To see (1) observe that in the graph G^J, a vertex $v \in X^J \cap B$ has at most $2k(k-1) - 1$ neighbors in B (as every vertex of $B \setminus Q$ has degree less than $2k(k-1)$ in G^B), while every vertex $v \in X^J \setminus B$ has at most k neighbors in B, as every component of $G' - B$ has at most k neighbors in B. Since $k \geqslant 2$ and thus $2k(k-1) - 1 \geqslant k$, this proves (1).

$$|Y^J| \leqslant (2k(k-1) + k + (2k(k-1))^2)|X^J| \leqslant 4k^4|X^J| \leqslant 4k^4\ell = \mathcal{O}(k^4 dt \log n). \tag{2}$$

To see (2), consider a dirty component $C \in \mathcal{C}$. Observe that either C contains a vertex of X^J or $N_{G'}(C) \cap V(G^J)$ contains a vertex of $N_{G^J}[X^J] \cap B$. There are at most $|X^J|$ dirty components of the first type, contributing in total at most $k|X^J|$ vertices to Y^J. For the dirty components of the second type, although there can be many of them, we observe that if $v \in N_{G'}(C) \cap N_{G^J}[X^J] \cap B$, then $N_{G'}(C) \cap V(G^J) \subseteq N_{G^B}[v]$. Hence, for every dirty component of the second type, it holds that $N_{G'}(C) \cap V(G^J) \subseteq N_{G^B}[N_{G^J}[X^J] \cap B]$. Since each vertex of $N_{G^J}[X^J] \cap B$ has degree less than $2k(k-1)$ in G^B, by (1) we have

$$\left|N_{G^B}\left[N_{G^J}[X^J] \cap B\right]\right| \leqslant (2k(k-1))^2|X^J|.$$

Adding the upper bound on $|N_{G^J}[X^J] \cap B|$ from (1), the bound (2) follows.

A component $C \in \mathcal{C}$ is *touched* if it is dirty or $N_{G'}(C)$ contains a vertex of Y^J. Let

$$Z^J := (N_{G^J}[Y^J] \cap B) \cup \bigcup_{C \in \mathcal{C}\colon C \text{ is touched}} N_{G'}(C) \cap V(G^J).$$

Now let us argue that

$$|Z^J| \leqslant 2k(k-1)|Y^J| \leqslant 8k^6|X^J| \leqslant 8k^6\ell = \mathcal{O}(k^6 dt \log n). \tag{3}$$

Indeed, if C is touched, then $N_{G'}(C)$ contains a vertex $v \in Y^J$ (if C is dirty, $N_{G'}(C) \cap V(G^J)$ is contained in Y^J), and then $N_{G'}(C)$ is contained in $N_{G^B}[v]$. Also, for $v \in Y^J$ we have $N_{G^J}[v] \cap B \subseteq N_{G^B}[v]$. Hence, $Z^J \subseteq N_{G^B}[Y^J]$. Since the maximum degree of a vertex of $B \setminus Q$ is $2k(k-1) - 1$, this proves (3).

For every touched $C \in \mathcal{C}$, denote $G_C := G^J[N_{G'}[C] \cap V(G^J)]$ and $T_C := ((T \cap C) \cup N_{G'}(C)) \cap V(G^J)$. We recurse on $(G_C, \mathbf{w}, T_C)$, obtaining $f_{G_C,\mathbf{w},T_C}$.

Let

$$G^Y := G^J - Y^J - \bigcup_{C \in \mathcal{C}\colon C \text{ is touched}} C.$$

Note that, by the definition of dirty and touched, G^Y is an induced subgraph of $G^J - N_{G^J}[X^J]$. Hence, (H^J, η^J) can be restricted to a (not necessarily rigid) extended strip decomposition $(H^J, \eta^{J,Y})$ of G^Y.

Let $T^Y := (T \cup Z^J) \cap V(G^Y)$. For every particle A of $(H^J, \eta^{J,Y})$, we recurse on $(G^Y[A], \mathbf{w}, T^Y \cap A)$, obtaining $f_{G^Y[A],\mathbf{w},T^Y \cap A}$. Then, we use these values with Lemma 7 to solve a Border MWIS instance $(G^Y, \mathbf{w}, T^Y)$, obtaining $f_{G^Y,\mathbf{w},T^Y}$.

Next, we iterate over every independent set $I_T \subseteq T^J \cup T^Y \cup Y^J$. For fixed I_T, let $I \subseteq V(G')$ be the union of the following sets:

(i) J,

(ii) I_T,

(iii) a maximum-weight independent set in G^Y whose intersection with T^Y is $I_T \cap Y^Y$,

(iv) for each touched component $C \in \mathcal{C}$, a maximum-weight independent set in G_C whose intersection with T_C is $I_T \cap N_{G'}[C]$.

Observe that I is independent. Indeed, all neighbors of J were removed when defining G^J, and Y^J separates touched components from each other and from G^Y. Thus, making sure that all partial solutions agree on Y^J, we ensure that I does not contain two adjacent vertices.

Second, note that the sets listed in (iii) and (iv) have weights, respectively, $f_{G^Y,\mathbf{w},T^Y}(I_T \cap T^Y)$ and $f_{G_C,\mathbf{w},T_C}(N_{G'}[C] \cap I_T)$. Consequently, I is an independent set in G', satisfying $I \cap (Q \cup T \cup T^Y \cup Y^J) = J \cup I_T$, of weight:

$$\mathbf{w}(J) + \mathbf{w}(I_T \setminus T^Y) + f_{G^Y,\mathbf{w},T^Y}(I_T \cap T^Y) + \sum_{\substack{C \in \mathcal{C} \\ C \text{ is touched}}} \left(f_{G_C,\mathbf{w},T_C}(N_{G'}[C] \cap I_T) - \mathbf{w}(I_T \cap N_{G'}(C))\right). \quad (4)$$

We update the cell $f_{G',\mathbf{w},T}((I_T \cup J) \cap T)$ with this value if it is larger than the previous value of this cell. This completes the description of the algorithm.

For the proof of correctness, we already observed that I, defined above, is an independent set and its weight is (4). Thus it suffices to argue that we can compute its weight using the information returned by appropriate recursive calls. This is indeed the case, as for every touched component C, the whole $N_{G'}(C) \cap V(G^J)$ is in the terminal set for the recursive call $(G_C, \mathbf{w}, T_C)$ and the whole $N_{G'}(C) \cap V(G^Y)$ is in Z^J and thus in the terminal set for the Border MWIS instance $(G^Y, \mathbf{w}, T^Y)$.

For the sake of analysis, consider a recursive call on $(G_C, \mathbf{w}, T_C)$ for a touched component C. If $|T| \leqslant 24k^6\ell$ and $U = V(G') \setminus T$, then $|T_C| \leqslant |T| + k \leqslant 32k^6\ell$ and $|V(G_C) \setminus T_C| \leqslant |C \setminus T| \leqslant |V(G') \setminus T|/2$. Otherwise, if $|T| > 24k^6\ell$ and $U = T$, then $|T_C| \leqslant |T|/2 + k \leqslant 16k^6\ell + k \leqslant 24k^6\ell$. Thus, the recursive call on $(G_C, \mathbf{w}, T_C)$ will fall under the first case of at most $24k^6\ell$ terminals.

Analogously, consider a recursive call on $(G^Y[A], \mathbf{w}, T^Y \cap A)$ for a particle A of $(H^J, \eta^{J,Y})$. If $|T| \leqslant 24k^6\ell$ and $U = V(G') \setminus T$, then $|T^Y \cap A| \leqslant |T^Y| \leqslant |T| + |Z^J| \leqslant 32k^6\ell$ due to (3). Furthermore, $|V(G^Y[A]) \setminus T^Y| \leqslant |V(G') \setminus T|/2$. Otherwise, if $|T| > 24k^6\ell$ and $U = T$, then $|T^Y \cap A| \leqslant |T|/2 + |Z^J| \leqslant 16k^6\ell + 8k^6\ell \leqslant 24k^6\ell$ again due to (3). Thus, the recursive call on $(G^Y[A], \mathbf{w}, T^Y \cap A)$ will fall under the first case of at most $24k^6\ell$ terminals.

Finally, note that a recursive call $(G', \mathbf{w}, T)$ without nonterminal vertices (i.e., with $T = V(G')$) is a leaf call.

We infer that all recursive calls satisfy the invariant of at most $32k^6\ell$ terminals and the depth of the recursion tree is bounded by $2\lceil \log n \rceil$ (as every second level the number of nonterminal vertices halves).

At each recursive call, we iterate over at most 2^k subsets $J \subseteq Q$. Lemma 8 ensures that the maximum degree of H^J is at most $2t-1$, while Lemma 9 ensures that every vertex of G^Y is used in at most $4t$ particles of $(H^J, \eta^{J,Y})$. In a subcall $(G_C, \mathbf{w}, T_C)$ for a touched component C, vertices of C are not used in any other call for the current choice of J, while all vertices of $V(G_C) \setminus C$ are terminals. Consequently, every nonterminal vertex v of G' is passed as a nonterminal vertex to a recursive subcall at most $2^k \cdot 4t$

number of times (and a terminal is always passed to a subcall as a terminal). Furthermore, a recursive call without nonterminal vertices is a leaf call. As the depth of the recursion is $\mathcal{O}(\log n)$, we infer that, summing over all recursive calls in the entire algorithm, the number of nonterminal vertices is bounded by $n^{\mathcal{O}(\log t+k)}$ and the total size of the recursion tree is $n^{\mathcal{O}(\log t+k)}$.

At each recursive call, we iterate over all 2^k subsets $J \subseteq Q$ and then we invoke Theorem 3 and iterate over all independent sets I_T in $T^J \cup T^Y \cup Y^J$. Thanks to the invariant $|T| \leqslant 32k^6\ell$ and bounds (2), and (3), this set is of size $\mathcal{O}(k^6\ell)$. Hence, every recursive call runs in time $n^{\mathcal{O}(k^6 t)+kc_{d,t}}$, where $c_{d,t}$ is a constant depending on d and t. As k is a constant depending on d, t, s, the final running time bound is polynomial in n. This completes the proof of Theorem 2.

5 Conclusion

While it is generally believed that MWIS is polynomial-time-solvable in $S_{t,t,t}$-free (and even $dS_{t,t,t}$-free) graphs (with no further assumptions), such a result seems currently out of reach. Thus it is interesting to investigate how further can we relax the assumptions on instances, as we did when going from Theorem 1 to Theorem 2. In particular, we used the assumption of K_r-freeness twice: once in Lemma 5 and then to argue that H (the pattern of an extended strip decomposition we obtain) is of bounded degree. On the other hand, the assumption of $K_{r,r}$-freeness was used just once: in Lemma 5. Thus it seems natural to try to prove the following conjecture.

Conjecture 10 *For every integers t, r there exists a polynomial-time algorithm that, given an $S_{t,t,t}$-free and K_r-free vertex-weighted graph $(G, \mathbf{w})$ computes the maximum possible weight of an independent set in $(G, \mathbf{w})$.*

Acknowledgements We acknowledge the welcoming and productive atmosphere at Dagstuhl Seminar 22481 "Vertex Partitioning in Graphs: From Structure to Algorithms," where some part of the work leading to the results in this paper was done.

References

[1] Tara Abrishami, Maria Chudnovsky, Cemil Dibek, and Paweł Rzążewski. Polynomial-time algorithm for maximum independent set in bounded-degree graphs with no long induced claws. In Niv Buchbinder Joseph (Seffi) Naor, editor, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference, January 9-12, 2022*, pages 1448–1470. SIAM, 2022. doi:10.1137/1.9781611977073.61.

[2] Tara Abrishami, Maria Chudnovsky, Cemil Dibek, Stephan Thomassé, Nicolas Trotignon, and Kristina Vušković. Graphs with polynomially many minimal separators. *Journal of Combinatorial Theory, Series B*, 152:248–280, 2021.

[3] Tara Abrishami, Maria Chudnovsky, Marcin Pilipczuk, and Paweł Rzążewski. Max weight independent set in sparse graphs with no long claws. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, pages 4:1–4:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.STACS.2024.4, doi:10.4230/LIPICS.STACS.2024.4.

[4] Vladimir 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] Vladimir E. Alekseev. On easy and hard hereditary classes of graphs with respect to the independent set problem. *Discret. Appl. Math.*, 132(1-3):17–26, 2003. doi:10.1016/S0166-218X(03)00387-1.

[6] Vladimir E. Alekseev. Polynomial algorithm for finding the largest independent sets in graphs without forks. *Discrete Applied Mathematics*, 135(1):3 – 16, 2004. Russian Translations II. URL: http://www.sciencedirect.com/science/article/pii/S0166218X02002901, doi:https://doi.org/10.1016/S0166-218X(02)00290-1.

[7] Gábor Bacsó, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Zsolt Tuza, and Erik Jan van Leeuwen. Subexponential-time algorithms for Maximum Independent Set in P_t-free and broom-free graphs. *Algorithmica*, 81(2):421–438, 2019. doi:10.1007/s00453-018-0479-5.

[8] Benjamin Bergougnoux, Tuukka Korhonen, and Igor Razgon. New width parameters for independent set: One-sided-mim-width and neighbor-depth. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers*, volume 14093 of *Lecture Notes in Computer Science*, pages 72–85. Springer, 2023. doi:10.1007/978-3-031-43380-1_6.

[9] Édouard Bonnet, Nicolas Bousquet, Stéphan Thomassé, and Rémi Watrigant. When maximum stable set can be solved in FPT time. In Pinyan Lu and Guochuan Zhang, editors, *30th International Symposium on Algorithms and Computation, ISAAC 2019, December 8-11, 2019, Shanghai University of Finance and Economics, Shanghai, China*, volume 149 of *LIPIcs*, pages 49:1–49:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ISAAC.2019.49.

[10] Andreas Brandstädt and Raffaele Mosca. Maximum weight independent set for ℓ-claw-free graphs in polynomial time. *Discret. Appl. Math.*, 237:57–64, 2018. doi:10.1016/j.dam.2017.11.029.

[11] Andreas Brandstädt and Raffaele Mosca. Maximum weight independent sets for (P_7, triangle)-free graphs in polynomial time. *Discret. Appl. Math.*, 236:57–65, 2018. doi:10.1016/j.dam.2017.10.003.

[12] Andreas Brandstädt and Raffaele Mosca. Maximum weight independent sets for ($S_{1,2,4}$,triangle)-free graphs in polynomial time. *Theor. Comput. Sci.*, 878-879:11–25, 2021. doi:10.1016/j.tcs.2021.05.027.

[13] Christoph Brause. A subexponential-time algorithm for the Maximum Independent Set Problem in P_t-free graphs. *Discret. Appl. Math.*, 231:113–118, 2017. doi:10.1016/j.dam.2016.06.016.

[14] Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michał Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016. doi:10.1137/15M1032077.

[15] Maria Chudnovsky, Marcin Pilipczuk, Michal Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in H-free graphs. *SIAM J. Comput.*, 53(1):47–86, 2024. URL: https://doi.org/10.1137/20m1333778, doi:10.1137/20M1333778.

[16] Maria Chudnovsky and Paul Seymour. Claw-free graphs. V. Global structure. *J. Combin. Theory Ser. B*, 98(6):1373–1410, 2008.

[17] Maria Chudnovsky, Stéphan Thomassé, Nicolas Trotignon, and Kristina Vušković. Maximum independent sets in (pyramid, even hole)-free graphs. *CoRR*, abs/1912.11246, 2019. URL: http://arxiv.org/abs/1912.11246, arXiv:1912.11246.

[18] D.G. Corneil, H. Lerchs, and L.Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163 – 174, 1981. URL: http://www.sciencedirect.com/science/article/pii/0166218X81900135, doi:https://doi.org/10.1016/0166-218X(81)90013-5.

[19] M. Cygan, F. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, 2015.

[20] Pavel Dvořák, Andreas Emil Feldmann, Ashutosh Rai, and Paweł Rzążewski. Parameterized inapproximability of Independent Set in H-free graphs. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science - 46th International Workshop, WG 2020, Leeds, UK, June 24-26, 2020, Revised Selected Papers*, volume 12301 of *Lecture Notes in Computer Science*, pages 40–53. Springer, 2020. doi:10.1007/978-3-030-60440-0_4.

[21] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.

[22] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976. URL: http://www.sciencedirect.com/science/article/pii/0304397576900591, doi:https://doi.org/10.1016/0304-3975(76)90059-1.

[23] Peter Gartland and Daniel Lokshtanov. Independent set on P_k-free graphs in quasi-polynomial time. In Sandy 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. doi:10.1109/FOCS46700.2020.00063.

[24] Peter Gartland, Daniel Lokshtanov, Tomás Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In Bojan Mohar, Igor Shinkar, and Ryan 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. doi:10.1145/3618260.3649791.

[25] Michael U. Gerber, Alain Hertz, and Vadim V. Lozin. Stable sets in two subclasses of banner-free graphs. *Discrete Applied Mathematics*, 132(1-3):121–136, 2003. doi:10.1016/S0166-218X(03)00395-0.

[26] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325–356. North-Holland, 1984. URL: https://www.sciencedirect.com/science/article/pii/S0304020808729438, doi:https://doi.org/10.1016/S0304-0208(08)72943-8.

[27] Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michał Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *ACM Trans. Algorithms*, 18(1):4:1–4:57, 2022. doi:10.1145/3414473.

[28] Ararat Harutyunyan, Michael Lampis, Vadim V. Lozin, and Jérôme Monnot. Maximum independent sets in subcubic graphs: New results. *Theor. Comput. Sci.*, 846:14–26, 2020. doi:10.1016/j.tcs.2020.09.010.

[29] Johan Håstad. Clique is hard to approximate within $n^{(1-\varepsilon)}$. In *Acta Mathematica*, pages 627–636, 1996.

[30] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972. doi:10.1007/978-1-4684-2001-2_9.

[31] Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 160–169. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.53.

[32] Subhash Khot and Ashok Kumar Ponnuswami. Better inapproximability results for Max Clique, Chromatic Number and Min-3Lin-Deletion. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 226–237, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[33] T. Kövari, V. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicae*, 3(1):50–57, 1954. URL: http://eudml.org/doc/210011.

[34] Ngoc Chi Lê, Christoph Brause, and Ingo Schiermeyer. The Maximum Independent Set Problem in subclasses of $S_{i,j,k}$-free graphs. *Electron. Notes Discret. Math.*, 49:43–49, 2015. doi:10.1016/j.endm.2015.06.008.

[35] Daniel Lokshantov, Martin Vatshelle, and Yngve Villanger. Independent set in P_5-free graphs in polynomial time. In Chandra 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. doi:10.1137/1.9781611973402.43.

[36] Vadim V. Lozin and Martin 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. doi:10.1016/j.jda.2008.04.001.

[37] Vadim V. Lozin, Martin Milanič, and Christopher Purcell. Graphs without large apples and the Maximum Weight Independent Set problem. *Graphs Comb.*, 30(2):395–410, 2014. doi:10.1007/s00373-012-1263-y.

[38] Vadim V. Lozin, Jérôme Monnot, and Bernard Ries. On the maximum independent set problem in subclasses of subcubic graphs. *J. Discrete Algorithms*, 31:104–112, 2015. doi:10.1016/j.jda.2014.08.005.

[39] Vadim V. Lozin and Dieter Rautenbach. Some results on graphs without long induced paths. *Inf. Process. Lett.*, 88(4):167–171, 2003. doi:10.1016/j.ipl.2003.07.004.

[40] Vadim V. Lozin and Igor Razgon. Tree-width dichotomy. *Eur. J. Comb.*, 103:103517, 2022. doi:10.1016/j.ejc.2022.103517.

[41] Frédéric Maffray and Lucas Pastor. Maximum weight stable set in (P_7, bull)-free graphs. *CoRR*, abs/1611.09663, 2016. URL: http://arxiv.org/abs/1611.09663.

[42] Konrad Majewski, Tomás Masarík, Jana Masaríková, Karolina Okrasa, Marcin Pilipczuk, Pawel Rzazewski, and Marek Sokolowski. Max weight independent set in graphs with no long claws: An analog of the gyárfás' path argument. *ACM Trans. Comput. Theory*, 16(2):8:1–8:18, 2024. doi:10.1145/3636422.

[43] George J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980. doi:https://doi.org/10.1016/0095-8956(80)90074-X.

[44] Raffaele Mosca. Stable sets in certain P_6-free graphs. *Discret. Appl. Math.*, 92(2-3):177–191, 1999. doi:10.1016/S0166-218X(99)00046-3.

[45] Raffaele Mosca. Stable sets of maximum weight in (P_7, banner)-free graphs. *Discrete Mathematics*, 308(1):20–33, 2008. doi:10.1016/j.disc.2007.03.044.

[46] Raffaele Mosca. Independent sets in (P_6, diamond)-free graphs. *Discret. Math. Theor. Comput. Sci.*, 11(1):125–140, 2009. doi:10.46298/dmtcs.473.

[47] Raffaele Mosca. Maximum weight independent sets in (P_6, co-banner)-free graphs. *Inf. Process. Lett.*, 113(3):89–93, 2013. doi:10.1016/j.ipl.2012.10.004.

[48] Raffaele Mosca. Independent sets in ($P_4 + P_4$, triangle)-free graphs. *Graphs Comb.*, 37(6):2173–2189, 2021. doi:10.1007/s00373-021-02340-7.

[49] Daishin Nakamura and Akihisa Tamura. A revision of Minty's algorithm for finding a maximum weight stable set of a claw-free graph. *Journal of the Operations Research Society of Japan*, 44(2):194–204, 2001. doi:10.15807/jorsj.44.194.

[50] Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Quasi-polynomial-time algorithm for independent set in P_t-free graphs via shrinking the space of induced paths. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 204–209. SIAM, 2021. doi:10.1137/1.9781611976496.23.

[51] Svatopluk Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15:307–309, 1974.

[52] Najiba Sbihi. Algorithme de recherche d'un stable de cardinalite maximum dans un graphe sans etoile. *Discrete Mathematics*, 29(1):53–76, 1980. doi:https://doi.org/10.1016/0012-365X(90)90287-R.

[53] Daniel Weißauer. On the block number of graphs. *SIAM J. Discret. Math.*, 33(1):346–357, 2019. doi:10.1137/17M1145306.

[54] Nikola Yolov. Minor-matching hypertree width. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 219–233. SIAM, 2018. doi:10.1137/1.9781611975031.16.

A Appendix: Proof of Lemma 5

For positive integers a, b, the Ramsey number of a and b, denoted by $\mathsf{Ram}(a, b)$, is the smallest integer r such that every graph on r vertices contains either an independent set of size a or a clique of size b. It is well-known that $\mathsf{Ram}(a, b) \leqslant \binom{a+b-2}{a-1}$.

For a graph H, a *subdivision of* H is any graph obtained from H by subdividing each edge arbitrarily many (possibly 0) times. By a *t-subdivision* (resp., *$(\leqslant t)$-subdivision*) we mean a subdivision where each edge was subdivided exactly (resp., at most t) times. A *proper subdivision* is one where each edge was subdivided at least once. By $S(r, t)$ we denote the t-subdivision of the r-leaf star. In particular, $S(3, t)$ is exactly $S_{t,t,t}$.

We will need the following two technical results shown by Lozin and Razgon [40].

Lemma 11 ([40, Lemma 2]) *There is a function $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with the following property. If a graph G contains a collection of $c(a, p)$ pairwise disjoint subsets of vertices, each of size at most a and with at least one edge between any two of them, then G contains $K_{p,p}$ as a subgraph.*

Lemma 12 ([40, Theorem 3]) *There is a function $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with the following property. Every graph G containing a $(\leqslant p)$-subdivision of $K_{m(h,p)}$ as a subgraph contains either $K_{p,p}$ as a subgraph or a proper $(\leqslant p)$-subdivision of $K_{h,h}$ as an induced subgraph.*

The following lemma is the main technical ingredient of the proof. It can be seen as a strengthening of Claim 2 in [40].

Lemma 13 *For all positive integers d, p, m there exists $k = k(d, p, m)$ such that the following holds. If G contains a k-block, then it contains:*

1. *a $(\leqslant p)$-subdivision of K_m as a subgraph or*
2. *$K_{p,p}$, as a subgraph, or*
3. *$dS_{p,p,p}$ as an induced subgraph.*

Proof. Let c be the function given by Lemma 11. Define the following constants

$$\begin{aligned} q &= 2\,\mathsf{Ram}(d, c(3p + 1, p)), \\ r &= (q/2 - 1)(3p + 1) + 3, \\ \ell &= \mathsf{Ram}(p, c(p, p)), \\ k' &= \max\left(m + q, p\binom{r}{2} + \ell\right), \\ k &= \mathsf{Ram}(2k', m). \end{aligned}$$

Suppose that G has a k-block $\widetilde{B}$ but no subgraph isomorphic to $K_{p,p}$ nor induced subgraph isomorphic to $dS_{p,p,p}$. We aim to show that G has a $(\leqslant p)$-subdivision of K_m as a subgraph. In particular, if $G[\widetilde{B}]$ contains K_m itself, we are done. Thus, by the choice of k, there is an independent set $B \subseteq \widetilde{B}$ of size $2k'$.

Consider a pair $\{x, y\}$ of distinct vertices from B. Since x and y are non-adjacent and cannot be separated by deleting fewer than k vertices, by Menger's theorem there is a set $\mathcal{P}(x, y)$ of $k \geqslant 2k'$ pairwise internally disjoint x-y-paths in G. Without loss of generality we can assume that each such path is induced.

Claim 1 *There is a subset $B' \subseteq B$ of size k' with the following property. For each pair of distinct vertices $x, y \in B'$ there is a set $\mathcal{P}'(x, y)$ of k' pairwise internally disjoint induced x-y-paths in G that do not contain any vertices from $B' \setminus \{x, y\}$.*

Proof of Claim. Let B' be any set of k' vertices from B. Consider any two distinct vertices $x, y \in B'$. As the interiors of paths in $\mathcal{P}(x, y)$ are pairwise disjoint, we observe that at most k' paths might contain vertices from $B' \setminus \{x, y\}$. Thus $\mathcal{P}(x, y)$ contains at least $|\mathcal{P}(x, y)| - k' \geqslant k'$ paths that do not intersect $B' \setminus \{x, y\}$. ⌟

Fix a pair $\{x, y\}$ of distinct vertices from B'. A path in $\mathcal{P}'(x, y)$ is *long* if it has at least p internal vertices and *short* otherwise. The pair $\{x, y\}$ is *distant* if at least ℓ paths in $\mathcal{P}'(x, y)$ are long.

Let $Q \subseteq B'$ be a minimum-size subset of B' that intersects all distant pairs. (It might be useful to think of Q as a minimum vertex cover in a graph with vertex set B' where the edges correspond to distant pairs.) We consider two cases, depending whether Q is "large" or "small".

Case 1 (large Q): $|Q| \geqslant q$. In this case there is a set consisting of at least $|Q|/2 \geqslant \mathsf{Ram}(d, c(3p+1, p))$ pairwise disjoint distant pairs (i.e., a matching in the mentioned graph). Let M be such a set of size exactly $\mathsf{Ram}(d, c(3p + 1, p))$.

Claim 2 *For each $\{x, y\} \in M$, there is a set $S^{\{x,y\}}$ contained in the union of paths in $\mathcal{P}'(x, y)$ that induces $S(r, p)$ in G.*

Proof of Claim. Fix $\{x, y\} \in M$. For a long path $P \in \mathcal{P}'(x, y)$, let $\mathsf{prefix}(P)$ be the set of p first vertices of P, starting from the side of x, but excluding x itself. So $\mathsf{prefix}(P)$ induces a p-vertex path, and $\mathsf{prefix}(P) \cup \{x\}$ induces a $(p + 1)$-vertex path in G.

Fix any set $\mathcal{P}'_{\text{long}}$ of ℓ long paths from $\mathcal{P}'(x, y)$, and define $Z = \{\mathsf{prefix}(P) \mid P \in \mathcal{P}'_{\text{long}}\}$. Consider an auxiliary graph $\mathbf{Z}$ with vertex set Z, in which two elements are adjacent if and only if there is an edge between one set and the other (we emphasize here that the sets in Z are pairwise disjoint).

If $\mathbf{Z}$ contains a clique of size $c(p, p)$, then, by Lemma 11, we obtain a subgraph isomorphic to $K_{p,p}$ in G, a contradiction. Thus, since $|Z| = \ell = \mathsf{Ram}(r, c(p, p))$, we obtain an independent set of size r in $\mathbf{Z}$. The p paths forming this independent set, together with x, induce the desired copy of $S(r, p)$. ⌟

Observe that even though the pairs in M are pairwise disjoint, the sets $S^{\{x,y\}}$ might still intersect. In the next claim we extract from them induced copies of $S_{p,p,p}$ that are pairwise disjoint.

Claim 3 *For each $\{x, y\} \in M$ there is an induced copy of $S_{p,p,p}$ contained in $S^{\{x,y\}}$, such that for any distinct $\{x, y\}, \{x', y'\} \in M'$ the corresponding copies of $S_{p,p,p}$ are disjoint.*

Proof of Claim. Fix an arbitrary order $\{x_1, y_1\}, \ldots, \{x_{|M|}, y_{|M|}\}$ on pairs in M. The proof is by induction.

The copy of $S_{p,p,p}$ for $\{x_1, y_1\}$ can be selected by picking any three out of r paths in the copy of $S(r, p)$ given by Claim 2. So let $i \in [1, |M| - 1]$ and suppose that for each $1 \leqslant j \leqslant i$ we have selected an induced copy S^j of $S_{p,p,p}$ contained in $S^{\{x_j,y_j\}}$. Note that in total we have selected $i \cdot (3p + 1)$ vertices. Consider the pair $\{x_{i+1}, y_{i+1}\}$ and let $P^1, \ldots, P^p$ be the t-vertex paths obtained from $S^{\{x_{i+1},y_{i+1}\}}$ by deleting x_{i+1}. Since each of the $i(3p + 1)$ selected vertices can belong to at most one of the paths $P^1, \ldots, P^p$, at most $i(3p+1) \leqslant (|M|-1)(3p+1)$ of these paths might intersect $S^1 \cup \ldots \cup S^i$. So, as $r = (|M|-1)(3p+1)+3$, there is always a choice of three paths that are disjoint with $S^1 \cup \ldots \cup S^i$. Recall that by Claim 1 the vertex x_{i+1} is not contained $\bigcup_{1 \leqslant j \leqslant i} S^j$. Thus the vertices of the three selected paths, together with x_{i+1}, form the set S^{i+1}. ⌟

Note that the copies of $S_{p,p,p}$ extracted in Claim 3, even though pairwise disjoint, might still have edges between them. Thus it remains to show that we can extract d copies of $S_{p,p,p}$ that together form an induced $dS_{p,p,p}$ in G. We proceed similarly as in the proof of Claim 2. Let S be the family consisting of induced copies of $S_{p,p,p}$ for all $\{x,y\} \in M$; they are given by Claim 3. Let $\mathbf{S}$ be the graph with vertex set S where two vertices are adjacent if and only if there is an edge from one set to another. Note that $|S| = \mathsf{Ram}(d, c(3p+1,p))$. An independent set in $\mathbf{S}$ of size d corresponds to an induced $dS_{p,p,p}$ in G. On the other hand, if $\mathbf{S}$ has a clique of size at least $c(3p+1,p)$, then by Lemma 11 we obtain $K_{p,p}$ as a subgraph of G. By the choice of $|S|$ one of these cases must happen, and both yield a contradiction. Thus we conclude that Case 1 cannot occur.

Case 2 (small Q): $Q < q$. Define $B'' = B' \setminus Q$; note that $|B''| \geqslant k' - q \geqslant m$. For any distinct $x, y \in B''$, let $\mathcal{P}''(x,y)$ be obtained from $\mathcal{P}'(x,y)$ by removing all long paths. By the definition of Q we observe that for all x,y we have $|\mathcal{P}''(x,y)| \geqslant k' - \ell \geqslant p\binom{m}{2}$.

Let R be any m-element subset of B''.

Claim 4 *For each pair $\{x,y\}$ of distinct vertices of R there is a path $P^{\{x,y\}} \in \mathcal{P}''(x,y)$ such the paths selected for distinct pairs are internally disjoint.*

Proof of Claim. The proof is similar to the proof of Claim 3. We use induction. Enumerate pairs of distinct vertices from R as $\{x_1,y_1\},\ldots,\{x_{\binom{m}{2}}, y_{\binom{m}{2}}\}$.

The path $P^{\{x_1,y_1\}}$ can be arbitrarily chosen from $\mathcal{P}''(x_1,y_1)$. Suppose that we have selected paths $P^{\{x_1,y_1\}},\ldots,P^{\{x_i,y_i\}}$ for some $1 \leqslant i < \binom{m}{2}$. Since each path is short, the selected paths have in total at most $i \cdot p < p\binom{m}{2}$ internal vertices.

Now consider the set $\mathcal{P}''(x_{i+1},y_{i+1})$. Recall that the paths in this set are pairwise internally disjoint. Since $|\mathcal{P}''(x_{i+1},y_{i+1})| \geqslant p\binom{m}{2}$, we observe that there is a path in $\mathcal{P}''(x_{i+1},y_{i+1})$ which is internally disjoint from all previously selected paths. We pick this path as $P^{\{x_{i+1},y_{i+1}\}}$. ⌟

Recall that for each distinct $x,y \in R$, the path $P^{\{x,y\}}$ does not contain vertices from $R \setminus \{x,y\}$. Thus the set R with the paths given by Claim 4 forms a $(\leqslant p)$-subdivision of K_m which is a subgraph of G. This concludes the proof of Lemma 13. □

Now we can proceed to the proof of Lemma 5.

Proof. (of Lemma 5.) Define

$$\begin{aligned} p &= \max(s,p), \\ h &= d(3t+1) \text{ (i.e., the number of vertices of } dS_{t,t,t}), \\ m &= m(h,p), \\ k &= k(d,p,m), \end{aligned}$$

where functions m and k are given, respectively, by Lemma 12 and Lemma 13.

For a contradiction, suppose that G has a k-block. As $K_{p,p}$ contains $K_{s,s}$ as a subgraph and $dS_{p,p,p}$ contains $dS_{t,t,t}$ as an induced subgraph, by Lemma 13 we conclude that G contains a $(\leqslant p)$-subdivision of K_m as a subgraph. By Lemma 12 we observe that G contains a proper $(\leqslant p)$-subdivision of $K_{h,h}$ as an induced subgraph.

It is straightforward to verify that this induced subgraph contains a subdivision of $dS_{t,t,t}$ (in fact, of any graph with at most h vertices and at most h edges) as an induced subgraph. Therefore G contains an induced subgraph isomorphic to $dS_{t,t,t}$, a contradiction. □

B Appendix: Proof of Lemma 7

Before we present the proof of Lemma 7, let us first recall how extended strip decompositions can be used to solve MWIS.

B.1 Independent sets in G and matchings

Let (H,η) be an extended strip decomposition of G. Section 3.3 of [15] shows a link between independent sets in G and matchings in an auxiliary graph H' (which is a slight modification of the host graph H). Assume for every particle A of (H,η) we have fixed an independent set $I(A)$ of $G[A]$ with weight $a(A)$.

The graph H' is created from H by adding, for every edge $xy \in E(H)$, a new vertex t_{xy} and edges $t_{xy}x$ and $t_{xy}y$. Furthermore, the weight function $\mathbf{w}' : E(H) \to \mathbb{Z}$ is defined as follows:

$$\begin{aligned}
\mathbf{w}'(t_{xy}x) &:= a(A_{xy}^{x}) - a(A_{xy}^{\perp}) - a(A_x),\\
\mathbf{w}'(t_{xy}y) &:= a(A_{xy}^{y}) - a(A_{xy}^{\perp}) - a(A_y),\\
\mathbf{w}'(xy) &:= a(A_{xy}^{xy}) - a(A_{xy}^{\perp}) - a(A_x) - a(A_y) - \sum_{z,\text{ s.t. } xyz \in T(H)} a(A_{xyz}).
\end{aligned}$$

Let

$$\mathcal{A}_0 = \{A_{xy}^{\perp} \mid xy \in E(H)\} \cup \{A_x \mid x \in V(H)\} \cup \{A_{xyz} \mid xyz \in T(H)\}.$$

and

$$a_0 = \sum_{A \in \mathcal{A}_0} a(A).$$

A *selection* is a function σ that assigns to every node $x \in V(H)$ an edge of H incident with x or a value $\perp$. Given a selection σ, we define a set $M(\sigma) \subseteq E(H')$ as follows:

- if $\sigma(x) = \perp$, then x is not an endpoint of any edge of $M(\sigma)$;
- if $\sigma(x) = xy$ but $\sigma(y) \neq xy$, then $xt_{xy} \in M(\sigma)$;
- if $\sigma(x) = xy = \sigma(y)$, then $xy \in M(\sigma)$.

Note that, for every selection σ, the set $M(\sigma)$ is a matching in H'. In the other direction, if M is a matching in H', then we define a selection $\sigma(M)$ as:

- if no edge of M is incident with $x \in V(H)$, then $\sigma(M)(x) = \perp$;
- if $t_{xy}x$ or xy belongs to M, then $\sigma(M)(x) = xy$.

It is straightforward to check that $M(\sigma(M)) = M$ for every matching M in H' and $\sigma(M(\sigma)) = \sigma$ for every selection σ. Hence, selections are in one-to-one correspondence with matchings in H'.

Let J be an independent set in G. Define a selection $\sigma(J)$ as follows: for every $x \in V(H)$, let $\sigma(J)(x) = \perp$ if $J \cap \bigcup_{y \in N_H(x)} \eta(xy, x) = \emptyset$ and otherwise $\sigma(J)(x) = xy$ for the unique edge xy incident with x with $J \cap \eta(xy, x) \neq \emptyset$. In the other direction, for a matching M in H', define a subset $\mathcal{A}(M)$ of particles of (H, η) as follows:

- start with $\mathcal{A}(M) := \mathcal{A}_0$;
- for every edge $t_{xy}x \in M$ for $x, y \in V(H)$, replace A_x and $A_{xy}^{\perp}$ with A_{xy}^{x};
- for every edge $xy \in M$ for $x, y \in V(H)$, replace A_x, A_y, $A_{xy}^{\perp}$, and all particles A_{xyz} for $z \in V(H), xyz \in T(H)$ with A_{xy}^{xy}.

Observe that the elements of $\mathcal{A}(M)$ are pairwise disjoint. Furthermore,

$$\bigcup \mathcal{A}(M) = V(G) \setminus \bigcup_{x \in V(H)} \bigcup_{\substack{y \in N_H(x)\\ y \neq \sigma(M)(x)}} \eta(xy, x). \tag{5}$$

We have the following.

Lemma 14 ([15]) *For every matching M in H', the set*

$$\bigcup_{A \in \mathcal{A}(M)} I(A)$$

is an independent set of weight

$$a_0 + \sum_{e \in M} \mathbf{w}'(e).$$

Lemma 15 ([15]) *For every independent set J in H', $\sigma(J)$ is a selection and*

$$\sum_{e \in M(\sigma(J))} \mathbf{w}'(e) = -a_0 + \sum_{A \in \mathcal{A}(M(\sigma(J)))} \left(a(A) - \mathbf{w}(A \cap J)\right).$$

Furthermore, for every $A \in \mathcal{A}(M(\sigma(J)))$, the set

$$(J \setminus A) \cup I(A)$$

is also an independent set in G.

B.2 Proof of Lemma 7

We iterate over every $I_T \subseteq T$. For fixed I_T, we aim at computing $f_{G,\mathbf{w},T}(I_T)$. If I_T is not independent, we set $f_{G,\mathbf{w},T}(I_T) = -\infty$. In the remainder of the proof, we show how to compute in polynomial time the value $f_{G,\mathbf{w},T}(I_T)$ for fixed independent $I_T \subseteq T$.

For a particle A of (H, η), let $a(A) := f_{G[A],\mathbf{w},T\cap A}(I_T \cap A)$ and let $I(A)$ be an independent set witnessing this value, that is, an independent set in $G[A]$ of weight $a(A)$ with $I(A) \cap T \cap A = I_T \cap A$. Note that as I_T is independent, the value $a(A)$ is not equal to $-\infty$ and such an independent set exists.

We say that $x \in V(H)$ is *forced* if $I_T \cap \bigcup_{y \in N_H(x)} \eta(xy, x) \neq \emptyset$. Note that since I_T is independent, if x is forced, then $\eta(xy, x) \cap I_T \neq \emptyset$ for exactly one edge xy incident with x. We call such an edge xy the *enforcer* of x. Note that an edge xy may be the enforcer of both x and y. We say that a selection σ is *compliant* if for every forced $x \in V(H)$, the value $\sigma(x)$ equals to the enforcer of x; by the equivalence of selections and matchings in H', we also say that a matching M in H' is compliant if $\sigma(M)$ is compliant.

We now observe that Lemmata 14 and 15 reduce our task to finding a compliant matching in H' of maximum possible weight.

Lemma 16 *If M is a compliant matching in H', then the set*

$$J := \bigcup_{A \in \mathcal{A}(M)} I(A)$$

is an independent set of weight

$$a_0 + \sum_{e \in M} \mathbf{w}'(e)$$

such that $J \cap T = I_T$.

Proof. Lemma 14 asserts that J is indeed an independent set in G with weight as in the statement. It remains to show that $J \cap T = I_T$.

First, for every $v \in J \cap T$, there exists a unique $A \in \mathcal{A}(M)$ such that $v \in A$. Then, $v \in I_T$ as $I(A) \cap A \cap T = A \cap I_T$. This proves $J \cap T \subseteq I_T$.

Second, we claim that for every $v \in I_T$ there exists $A \in \mathcal{A}(M)$ with $v \in A$. This is immediate from (5) unless $v \in \eta(xy, x)$ for some $xy \in E(H)$. However, in this case, as M is compliant, $\sigma(M)(x) = xy$ and either A^x_{xy} or A^{xy}_{xy} belongs to $\mathcal{A}(M)$. Hence, as $v \in I_T \cap A$ and $I(A) \cap A \cap T = A \cap I_T$, we have $v \in I(A) \subseteq J$. This proves $I_T \subseteq J \cap T$ and completes the proof of the lemma. □

Lemma 17 *If J is an independent set in G with $J \cap T = I_T$, then $M(\sigma(J))$ is a compliant matching with*

$$\sum_{e \in M(\sigma(J))} \mathbf{w}'(e) \geqslant -a_0 + \mathbf{w}(J).$$

Proof. By Lemma 15, $\sigma(J)$ is a selection. Furthermore, for every $A \in \mathcal{A}(M(\sigma(J)))$ we have $J \cap A \cap T = I_T \cap A$. By the optimality of $I(A)$, $a(A) \geqslant \mathbf{w}(J \cap A)$. Consequently, Lemma 15 implies the promised bound on the weight of $M(\sigma(J))$.

It remains to check that $M(\sigma(J))$ is compliant. This is immediate: if xy is the enforcer of $x \in V(H)$, then by definition $\eta(xy, x) \cap I_T \neq \emptyset$, so $\eta(xy, x) \cap J \neq \emptyset$ and consequently $\sigma(J)(x) = xy$. This completes the proof of the lemma. □

Hence, it remains to show how to find a maximum-weight compliant matching in H'.

Lemma 18 *A maximum-weight compliant matching in $(H', \mathbf{w}')$ can be found in polynomial time.*

Proof. Construct a family $\mathcal{C}$ as follows: for every forced x with enforcer xy, add the set $\{t_{xy}x, xy\} \subseteq E(H')$ to $\mathcal{C}$. We seek for a matching M' in H' of maximum weight among matchings such that for every $C \in \mathcal{C}$ it holds that $|C \cap M| = 1$.

Execute the following reduction rules. First, for forced x with enforcer xy, delete from H' and all sets in $\mathcal{C}$ the edge $t_{xy}y$ and all edges incident with x except for xy and $t_{xy}x$ (these edges never belong to the matching we are looking for). Then, exhaustively, as long as there exists $C \in \mathcal{C}$ of size 1, pick the unique element e of C and include it into the solution: delete from $\mathcal{C}$ all sets containing e and delete all edges incident with an endpoint of e from H' and all remaining elements of $\mathcal{C}$. Let M_0 be the set of edges e on which the last reduction rule was executed in the course of the algorithm.

Let $C = \{t_{xy}x, xy\}$ be an element that remains in $\mathcal{C}$. Observe that now in H' the vertex t_{xy} is of degree 1 and x is of degree 2 and, furthermore, y is not forced. The sought matching M needs to take one of the edges of C; we gadget out this choice by modifying the weight of xy to $\mathbf{w}'(xy) - \mathbf{w}'(t_{xy}x)$ and deleting the edge $t_{xy}x$. Let M_1 be the set of edges $t_{xy}x$ that were deleted in this process and let $(H'', \mathbf{w}'')$ be the final graph.

For a matching M'' in H'', observe that a set M consisting of M_0, M'' and an edge $t_{xy}x \in M_1$ whenever $xy \notin M''$, is a compliant matching in H' of weight $\mathbf{w}'(M) = \mathbf{w}''(M'') + \mathbf{w}'(M_0) + \mathbf{w}'(M_1)$. In the other direction, for a compliant matching M in H', observe that $M_0 \subseteq M$ and, furthermore, $M'' := M \cap E(H'')$ is a matching in H'' of weight $\mathbf{w}''(M'') = \mathbf{w}'(M) - \mathbf{w}'(M_0) - \mathbf{w}'(M_1)$ as for every $t_{xy}x \in M_1$, either $t_{xy}x \in M$ and $xy \notin M$ (and then $t_{xy}x$ contributes $\mathbf{w}'(t_{xy}x)$ to $\mathbf{w}'(M_1)$ and $\mathbf{w}'(M)$ and nothing to $\mathbf{w}''(M'')$) or $xy \in M$ and $t_{xy}x \notin M$ (and then xy contributes $\mathbf{w}'(xy)$ to $\mathbf{w}'(M)$, $\mathbf{w}'(xy) - \mathbf{w}'(t_{xy}x)$ to $\mathbf{w}''(M'')$ and $t_{xy}x$ contributes $\mathbf{w}'(t_{xy}x)$ to $\mathbf{w}'(M_1)$).

Consequently, we reduced the problem of finding a maximum-weight compliant matching in $(H', \mathbf{w}')$ to a classic maximum-weight matching problem in $(H'', \mathbf{w}'')$, which is solvable in polynomial time [21]. This completes the proof of the lemma and of Lemma 7. ☐