Max Weight Independent Set in sparse graphs with no long claws

Tara Abrishami
Department of Mathematics, University of Hamburg

Maria Chudnovsky
Princeton University

Marcin Pilipczuk
Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Paweł Rzążewski
Faculty of Mathematics and Information Science, Warsaw University of Technology
Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Abstract

We revisit the recent polynomial-time algorithm for the Max Weight Independent Set (MWIS) problem in bounded-degree graphs that do not contain a fixed graph whose every component is a subdivided claw as an induced subgraph [Abrishami, Chudnovsky, Dibek, Rzążewski, SODA 2022].

First, we show that with an arguably simpler approach we can obtain a faster algorithm with running time $n^{\mathcal{O}(\Delta^2)}$, where n is the number of vertices of the instance and Δ is the maximum degree. Then we combine our technique with known results concerning tree decompositions and provide a polynomial-time algorithm for MWIS in graphs excluding a fixed graph whose every component is a subdivided claw as an induced subgraph, and a fixed biclique as a subgraph.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Max Weight Independent Set, subdivided claw, hereditary classes

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.21

Related Version *Extended version*: `https://arxiv.org/abs/2309.16995` [2]

Funding *Tara Abrishami*: Supported by the National Science Foundation Award Number DMS-2303251 and the Alexander von Humboldt Foundation.
Maria Chudnovsky: Supported by NSF-EPSRC Grant DMS-2120644 and by AFOSR grant FA9550-22-1-008.
Marcin Pilipczuk: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.
Paweł Rzążewski: 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)

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

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. 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 $(G, \mathbf{w})$. MWIS is one of canonical hard problems – it is NP-hard [23], W[1]-hard [15], hard to approximate [22]. Thus a very natural research direction is to consider restricted instances and try to capture the boundary between "easy" and "hard" cases.

State of the art. The study of complexity of MWIS in restricted graph classes is a central topic in algorithmic graph theory [20, 45, 27, 19, 14, 4]. A 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 this paper we are interested in graph classes defined by forbidding certain induced trees. Let $\mathcal{S}$ be the family of subcubic trees H with at most one vertex of degree 3. In other words, every such H is either a path or a subdivision of the *claw*: the three-leaf star. For integers $a, b, c \geqslant 1$, by $S_{a,b,c}$ we denote the claw whose edges were subdivided, respectively, $a-1$, $b-1$, and $c-1$ times.

As already observed by Alekseev in the early 1980s [3], if H is connected, MWIS remains NP-hard in H-free graphs unless $H \in \mathcal{S}$. In the past few years we have witnessed rapid progress in development of algorithms for MWIS in these remaining cases [6, 10, 16, 42, 13, 17, 35]. In particular, it is known that for each $H \in \mathcal{S}$, the MWIS problem can be solved in quasipolynomial time in H-free graphs [16, 42, 17]. This is a strong indication that the problem is not NP-hard. However, we are still very far from obtaining polynomial-time algorithms. Such results are known only for very small forbidden paths [27, 19], subdivided claws [36, 43, 5, 28], or their disjoint unions [31, 7]. There is also 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. [26, 8, 18, 32, 34, 38, 29, 30, 21, 37, 39, 40, 41, 9].

In a somewhat perpendicular direction, Abrishami, Chudnovsky, Dibek, and Rzążewski [1] proved that for every $H \in \mathcal{S}$, the MWIS problem can be solved in polynomial time in H-free graphs of bounded degree (where the degre of the polynomial depends on H, and the degree bound). Let us remark that the algorithm of [1] is very technical, and the dependency of the complexity on the degree bound is involved.

Our results. As a warm-up, we present a polynomial-time algorithm for H-free graphs of bounded degree, where $H \in \mathcal{S}$. It is significantly simpler than the one by Abrishami et al. [1] and has much better dependency on the maximum degree.

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

Note that by picking appropriate 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 $S_{t,t,t}$-free graphs, even if instead of bounding the maximum degree, we forbid a fixed biclique *as a subgraph*.

▶ **Theorem 2.** *For every fixed integer t, and s there exists a polynomial-time algorithm that, given a vertex-weighted graph $(G, \mathbf{w})$ that does not contain $S_{t,t,t}$ 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 Kővári-Sós-Turán theorem [25], classes that exclude $K_{s,s}$ as a subgraph capture all hereditary classes of *sparse graphs*, where by "sparse" we mean "where every graph has 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 [1, 13, 17], the crucial tool in handling $S_{t,t,t}$-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 [35].

▶ **Theorem 3** ([35, 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 an integer t, in polynomial time outputs either:*

- *an induced copy of $S_{t,t,t}$ in G, or*
- *a set X of size at most $(11 \log n + 6)(t + 1)$ 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 [35, 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 [17].

Consider the setting of Theorem 1, i.e., the graph G has maximum degree Δ. Apply Theorem 3 to G with $U = V(G)$. If we get the first outcome, i.e., an induced $S_{t,t,t}$ in G, we return it and terminate. So assume that we get the second outcome, i.e., the set X. Note that as $|X| = \mathcal{O}(t \log n)$, we have $|N[X]| = \mathcal{O}(t\Delta \log n)$. 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 $n^{\mathcal{O}(t\Delta \log n)}$, as the branching step yields up to $2^{|N[X]|} = n^{\mathcal{O}(t\Delta)}$ 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
$f_{G,\mathbf{w},T}(I_T) = \max\{\mathbf{w}(I) \mid I \subseteq V(G) \text{ and } I \text{ is independent, and } I \cap T = I_T\}$.

A similar application of a border version of the problem to postpone branching in recursion appeared for example in the technique of recursive understanding [24, 11].

Let us return to our setting, where we have a set X of size $\mathcal{O}(t \log n)$ 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 $\mathcal{O}(t\Delta^2 \log n)$.

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 $\mathcal{O}(t\Delta^2 \log n)$ 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 $\mathcal{O}(t\Delta^2 \log n)$ 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 $n^{\mathcal{O}(t\Delta^2)}$, which is within the promised time bound.

We remark that this approach is arguably significantly simpler and more direct than the decomposition techniques used in [1] and, furthermore, results in a much better dependency on Δ in the exponent in the final running time bound.

Let us now move to the more general setting of Theorem 2. Here, the starting point are recent results of Weißauer [44] and Lozin and Razgon [33] 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 is a set of k vertices, no two of which can be separated by deleting fewer than k vertices. The following result was shown by Weißauer (we refer to preliminaries for standard definitions of tree decompositions and torsos).

▶ **Theorem 4** (Weißauer [44]). *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 adhesion less than k, in which every torso has at most k vertices of degree larger than $2k(k-1)$.*

Even though the statement of the result in [44] is just existential, the proof actually yields a polynomial-time algorithm to compute such a tree decomposition.

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

▶ **Lemma 5** (Lozin and Razgon [33]). *For any t and s there exists k such that the following holds. Every $S_{t,t,t}$-free graph with no subgraph isomorphic to $K_{s,s}$ has no k-block.*

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

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

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, one requires some care to combine this with the information passed over adhesions in 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. 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 [35, 17]. For a graph H, by $T(H)$ we denote the set of triangles in H. 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 $xyz \in T(H)$,

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 $\eta(xz, x)$.
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 $xyz \in T(H)$.

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 proofs of analogous statements of [1, 13] and is included for completeness in Appendix A.

▶ **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 t be a positive integer 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

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

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 $\ell = \mathcal{O}(t \log n)$, this step takes $n^{\mathcal{O}(t\Delta^2)}$ 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 $S_{t,t,t}$ 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 $|T^*| \leqslant 5\Delta^4\ell$. For every particle A of (H, η), 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 $S_{t,t,t}$, which can be directly returned). Use Lemma 7 to obtain a solution $f_{G^*,\mathbf{w},T^*}$ to Border MWIS instance $(G^*, \mathbf{w}, T^*)$.

Finally, 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, $n^{\mathcal{O}(t\Delta^2)}$ time to compute $f_{G^*,\mathbf{w},T^*}$ using Lemma 7, and $n^{\mathcal{O}(t\Delta^2)}$ 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 $n^{\mathcal{O}(t\Delta^2)}$.

At every non-leaf recursive call, we make subcalls to $(G^*_A, \mathbf{w}, T^*_A)$ for every particle A of (H, η). Lemmata 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 $n^{\mathcal{O}(t\Delta^2)}$.

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 t be a positive integer and let k be the constant depending on t from Corollary 6. Again, let $(G, \mathbf{w})$ be the input vertex-weighted graph, let $n := |V(G)|$, and let

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

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

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 $32k^5\ell$ 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 $I_T \subseteq T$.

If $|V(G')| \leqslant 32k^5\ell$, 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 $\ell = \mathcal{O}(t \log n)$ and k is a constant depending on t and s, this step takes polynomial time. This completes the description of the leaf step of the recursion.

Otherwise, if $|V(G')| > 32k^5\ell$, we invoke Corollary 6 on G', obtaining a tree decomposition $(\mathcal{T}, \beta)$ of G'. If $|T| \leqslant 24k^5\ell$, let $U := V(G') \setminus T$, and otherwise, let $U := T$.

For every $t_1t_2 \in E(\mathcal{T})$, 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. 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 $|N_{G'}(C)| < k$ (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 larger than $2k(k-1)$. Let Q be the set of vertices of G^B of degree larger than $2k(k-1)$.

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 an 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

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

The following bounds will be important for further steps.

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

To see (1) observe that a vertex $v \in X^J \cap B$ has at most $2k(k-1)$ neighbors in B (as every vertex of $B \setminus Q$ has degree at most $2k(k-1)$ 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. This proves (1).

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

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 the degree of each vertex of G_B is at most $2k(k-1)$, by (1) we have

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

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

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

Using similar arguments as before, we can prove

$$|Z^J| \leqslant (2k(k-1)+1)|Y^J| \leqslant 8k^5|X^J| \leqslant 8k^5\ell = \mathcal{O}(k^5 t \log n). \quad (3)$$

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 $2k(k-1)$, this proves (3).

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

Let

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

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 $(H^J, \eta^{J,Y})$, recurse on $(G^Y[A], \mathbf{w}, T^Y \cap A)$, obtaining $f_{G^Y[A],\mathbf{w},T^Y \cap A}$. Then, use these values with Lemma 7 to solve a Border MWIS instance $(G^Y, \mathbf{w}, T^Y)$, obtaining $f_{G^Y,\mathbf{w},T^Y}$.

Iterate over every independent set $I_T \subseteq (T \cap V(G^J)) \cup T^Y \cup Y^J$. Observe that G' admits an independent set I with $I \cap (Q \cup T \cup T^Y \cup Y^J) = J \cup I_T$ and weight:

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

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 finishes the description of the algorithm.

For correctness, it suffices to note that 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 $|T| \leqslant 24k^5\ell$ and $U = V(G')\backslash T$, then $|T_C| \leqslant |T|+k \leqslant 32k^5\ell$ and $|V(G_C)\backslash T_C| \leqslant |C\backslash T| \leqslant |V(G')\backslash T|/2$. Otherwise, if $|T| > 24k^5\ell$ and $U = T$, then $|T_C| \leqslant |T|/2+k \leqslant 16k^5\ell+k \leqslant 24k^5\ell$. Thus, the recursive call on $(G_C, \mathbf{w}, T_C)$ will fall under the first case of at most $24k^5\ell$ 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 $|T| \leqslant 24k^5\ell$ and $U = V(G') \setminus T$, then $|T^Y \cap A| \leqslant |T^Y| \leqslant |T| + |Z^J| \leqslant 32k^5\ell$ due to (3). Furthermore, $|V(G^Y[A]) \setminus T^Y| \leqslant |V(G') \setminus T|/2$. Otherwise, if $|T| > 24k^5\ell$ and $U = T$, then $|T^Y \cap A| \leqslant |T|/2 + |Z^J| \leqslant 16k^5\ell + 8k^5\ell \leqslant 24k^5\ell$ 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 $24k^5\ell$ 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 $32k^5\ell$ 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 $(T \cap V(G^J)) \cup T^Y \cup Y^J$. Thanks to the invariant $|T| \leqslant 32k^5\ell$ and bounds (2), and (3), this set is of size $\mathcal{O}(k^5\ell)$. Hence, every recursive call runs in time $n^{\mathcal{O}(k^5 t) + k \cdot c_t}$, where c_t is a constant depending on t. As k is a constant depending on t and s, the final running time bound is polynomial.

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 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 $(G, \mathbf{w})$.*

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, Marcin Pilipczuk, and Paweł Rzążewski. Max Weight Independent Set in sparse graphs with no long claws. *CoRR*, abs/2309.16995, 2023. URL: https://doi.org/10.48550/arXiv.2309.16995, arXiv:2309.16995, doi:10.48550/ARXIV.2309.16995.

3 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.

4 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.

5 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:10.1016/S0166-218X(02)00290-1.

6 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.

7 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.

8 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.

9 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.

10 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.

11 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.

12 Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs. *CoRR*, abs/1907.04585, 2019. arXiv:1907.04585.

13 Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2260–2278. SIAM, 2020. doi:10.1137/1.9781611975994.139.

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

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

16 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.

17 Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. *CoRR*, abs/2305.15738, 2023. arXiv:2305.15738, doi:10.48550/arXiv.2305.15738.

18 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.

19 Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michał Pilipczuk. Polynomial-time algorithm for Maximum Weight Independent Set on P_6-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1257–1271. SIAM, 2019. doi:10.1137/1.9781611975482.77.

20 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:10.1016/S0304-0208(08)72943-8.

21 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.

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

23 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.

24 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.

25 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.

26 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.

27 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.

28 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.

29 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.

30 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.

31 Vadim V. Lozin and Raffaele Mosca. Independent sets in extensions of $2K_2$-free graphs. *Discret. Appl. Math.*, 146(1):74–80, 2005. doi:10.1016/j.dam.2004.07.006.

32 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.

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

34 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`.

35 Konrad Majewski, Tomáš Masařík, Jana Novotná, Karolina Okrasa, Marcin Pilipczuk, Paweł Rzążewski, and Marek Sokołowski. Max Weight Independent Set in Graphs with No Long Claws: An Analog of the Gyárfás' Path Argument. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 93:1–93:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2022.93`.

36 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:10.1016/0095-8956(80)90074-X`.

37 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`.

38 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`.

39 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`.

40 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`.

41 Raffaele Mosca. Independent sets in (P_{4+4}, triangle)-free graphs. *Graphs Comb.*, 37(6):2173–2189, 2021. `doi:10.1007/s00373-021-02340-7`.

42 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`.

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

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

45 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 7

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 $\mathbf{w}(a(A))$ 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.

The arguments now follow very closely the outline of Section 3.3 of [12].

We construct a set $\mathcal{P}$ of particles and an edge-weighted graph $(H', \mathbf{w}')$ as follows. We start with $\mathcal{P} = \emptyset$, $V(H') = V(H)$, and $E(H') = \emptyset$.

For every $x \in V(H)$ that is not forced, add A_x to $\mathcal{P}$. For every $xyz \in T(H)$ such that neither of the edges xy, yz, or xz is the enforcer of *both* its endpoints, add A_{xyz} to $\mathcal{P}$. For every $e = xy \in E(H)$, proceed as follows.

1. If neither x nor y is forced, we add to H' a new vertex t_e and edges $t_e x$, $t_e y$, xy, and set the edge weights $\mathbf{w}'$ as follows:

$$\mathbf{w}'(t_e x) := a(A^x_{xy}) - a(A^\perp_{xy}) - a(A_x),$$
$$\mathbf{w}'(t_e y) := a(A^y_{xy}) - a(A^\perp_{xy}) - a(A_y),$$
$$\mathbf{w}'(xy) := a(A^{xy}_{xy}) - a(A^\perp_{xy}) - a(A_x) - a(A_y) - \sum_{z, \text{ s.t. } xyz \in T(H)} a(A_{xyz}).$$

   Furthermore, add $A^\perp_{xy}$ to $\mathcal{P}$.
2. If exactly one of x and y is forced, say w.l.o.g. x is forced and y is not forced, proceed as follows.
   a. If xy is the enforcer of x, then add to H' an edge xy with weight

$$\mathbf{w}'(xy) := a(A^{xy}_{xy}) - a(A^x_{xy}) - a(A_y) - \sum_{z, \text{ s.t. } xyz \in T(H)} a(A_{xyz}).$$

   Furthermore, add A^x_{xy} to $\mathcal{P}$.
   b. If xy is not the enforcer of x, then add to H' a new vertex t_e and an edge $t_e y$ with weight

$$\mathbf{w}'(t_e y) := a(A^y_{xy}) - a(A^\perp_{xy}) - a(A_y).$$

   Furthermore, add $A^\perp_{xy}$ to $\mathcal{P}$.
3. If both x and y are forced, proceed as follows.
   a. If xy is neither the enforcer of x nor of y, add $A^\perp_{xy}$ to $\mathcal{P}$.
   b. If xy is the enforcer of x but not of y add A^x_{xy} to $\mathcal{P}$.
   c. If xy is the enforcer of y but not of x add A^y_{xy} to $\mathcal{P}$.
   d. If xy is the enforcer of both x and y, add A^{xy}_{xy} to $\mathcal{P}$.

This finishes the description of the construction of $\mathcal{P}$ and $(H', \mathbf{w}')$. In the next two paragraphs we make two observations that follow by a direct check from the definitions.

Observe that $I_0 := \bigcup_{A \in \mathcal{P}} I(A)$ is independent in G and has weight $a_0 := \sum_{A \in \mathcal{P}} a(A)$. Furthermore, for every $A \in \mathcal{P}$, we have $I_0 \cap A \cap T = I_T \cap A$ and $I_0 \cap T = I_T$. We think of I_0 as the "base" solution for $f_{G,\mathbf{w},T}(I_T)$.

Observe also that all weights $\mathbf{w}'$ of H' are nonnegative, as A^x_{xy} contains both $A^\perp_{xy}$ and A_x while A^{xy}_{xy} contains $A^\perp_{xy}$, A_x, A_y, as well as all A_{xyz} for all triangles xyz containing the edge xy.

We will be asking for a maximum weight matching in $(H', \mathbf{w}')$. Intuitively, taking an edge $t_e x$ to such a matching corresponds to replacing in I_0 the parts $I(A^\perp_{xy})$ and $I(A_x)$ with the part $I(A^x_{xy})$ while taking an edge xy to such a matching corresponds to replacing in I_0 the parts $I(A^\perp_{xy})$, $I(A_x)$, $I(A_y)$ and all parts $I(A_{xyz})$ for triangles xyz containing the edge xy with part $I(A^{xy}_{xy})$.

From another perspective, fix $x \in V(H)$ and recall that the sets $\eta(xy, x)$ for $y \in N_H(x)$ are complete to each other. Hence, any independent set in G can contain vertices in at most one of such sets. For an edge $e = xy \in E(H)$, taking an edge xy or $t_e x$ in a matching in

H' corresponds to choosing that, among all neighbors of x in H, the neighbor y is such that the set $\eta(xy, x)$ is allowed to contain vertices of the sought independent set. (Choosing $xy \in E(H')$ to the matching corresponds to allowing both $\eta(xy, x)$ and $\eta(yx, y)$ to contain vertices of the sought independent set.)

However, there is a delicacy if I_T contains a vertex of some interface $\eta(xy, x)$. Then, in some sense I_T already forces some choices in the corresponding matching in H'. This is modeled above by having alternate construction for vertices $x \in V(H)$ that are forced.

The following two claims prove that $f_{G,\mathbf{w},T}(I_T)$ equals a_0 plus the maximum possible weight of a matching in $(H', \mathbf{w}')$ and thus complete the proof of Lemma 7. Their proofs follow exactly the lines of the proofs of Claims 3.7 and 3.8 of Section 3.3 of [12] and are thus omitted.

▷ Claim 11. Let I be an independent set in G with $I \cap T = I_T$. Let M be the set of edges of H' defined as follows: for every $e = xy \in E(H)$, if $\eta(xy, x) \cap I \neq \emptyset$ and $\eta(xy, y) \cap I \neq \emptyset$, then $xy \in M$, if $\eta(xy, x) \cap I \neq \emptyset$ and $\eta(xy, y) \cap I = \emptyset$, then $t_e x \in M$, and if $\eta(xy, x) \cap I = \emptyset$ and $\eta(xy, y) \cap I \neq \emptyset$, then $t_e y \in M$. Then, all the above edges indeed exist in H' and M is a matching. Furthermore, the weight of I is at most $a_0 + \sum_{e \in M} \mathbf{w}'(e)$.

▷ Claim 12. Let M be a matching in H'. Let $\mathcal{P}_M$ be the set of particles of (H, η) defined as follows. Start with $\mathcal{P}_M := \mathcal{P}$ and then for every edge $e = xy \in E(H)$,

- if $xy \in M$, insert A_{xy}^{xy} into $\mathcal{P}_M$ and remove from $\mathcal{P}_M$ the following particles if present: A_{xy}^{x}, A_{xy}^{y}, $A_{xy}^{\perp}$, A_x, A_y, A_{xyz} for any $z \in V(H)$ such that $xyz \in T(H)$.
- if $t_e x \in M$ (resp. $t_e y \in M$), insert A_{xy}^{x} (resp. A_{xy}^{y}) into $\mathcal{P}_M$, and remove from $\mathcal{P}_M$ the following particles if present: $A_{xy}^{\perp}$ and A_x (resp. A_y).

Then $I_M := \bigcup_{A \in \mathcal{P}_M}$ is an independent set in G with $I_M \cap T = I_T$ and of weight at least $a_0 + \sum_{e \in M} \mathbf{w}'(e)$.