Maximum Weight Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time*

Peter Gartland
University of California
Santa Barbara, USA
petergartland@ucsb.edu

Daniel Lokshtanov
University of California
Santa Barbara, USA
daniello@ucsb.edu

Tomáš Masařík
University of Warsaw
Warsaw, Poland
masarik@mimuw.edu.pl

Marcin Pilipczuk
University of Warsaw
Warsaw, Poland
IT University
Copenhagen, Denmark
malcin@mimuw.edu.pl

Michał Pilipczuk
University of Warsaw
Warsaw, Poland
michal.pilipczuk@mimuw.edu.pl

Paweł Rzążewski
University of Warsaw
Warsaw, Poland
Warsaw University of Technology
Warsaw, Poland
p.rzazewski@mini.pw.edu.pl

ABSTRACT

We show that the Maximum Weight Independent Set problem (MWIS) can be solved in quasi-polynomial time on H-free graphs (graphs excluding a fixed graph H as an induced subgraph) for every H whose every connected component is a path or a subdivided claw (i.e., a tree with at most three leaves). This completes the dichotomy of the complexity of MWIS in $\mathcal{F}$-free graphs for any finite set $\mathcal{F}$ of graphs into NP-hard cases and cases solvable in quasi-polynomial time, and corroborates the conjecture that the cases not known to be NP-hard are actually polynomial-time solvable.

The key graph-theoretic ingredient in our result is as follows. Fix an integer $t \geqslant 1$. Let $S_{t,t,t}$ be the graph created from three paths on t edges by identifying one endpoint of each path into a single vertex. We show that, given a graph G, one can in polynomial time find either an induced $S_{t,t,t}$ in G, or a balanced separator consisting of $O(\log |V(G)|)$ vertex neighborhoods in G, or an extended strip decomposition of G (a decomposition almost as useful for recursion for MWIS as a partition into connected components) with each particle of weight multiplicatively smaller than the weight of G. This is a strengthening of a result of Majewski, Masařík, Novotná, Okrasa, Pilipczuk, Rzążewski, and Sokołowski [Transactions on Computation Theory 2024] which provided such an extended strip decomposition only after the deletion of $O(\log |V(G)|)$ vertex neighborhoods. To reach the final result, we employ an involved branching strategy that relies on the structural lemma presented above.

CCS CONCEPTS

• **Theory of computation** → **Graph algorithms analysis**; • **Mathematics of computing** → **Graph theory**.

KEYWORDS

Max independent set, subdivided claw, quasipolynomial algorithm.

ACM Reference Format:
Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. 2024. Maximum Weight Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3618260.3649791

*The full version of the paper including the omitted proofs is available on arXiv [20].

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649791

1 INTRODUCTION

The Maximum Weight Independent Set (MWIS) problem takes as input a graph G with vertex weights $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geqslant 0}$ and asks for a set $X \subseteq V(G)$ of maximum possible weight that is *independent* (sometimes also called *stable*): no two vertices of X are adjacent. This classic combinatorial problem plays an important role as a central hard problem in several areas of computational complexity: it appears as one of the NP-hard problems on the celebrated list of Karp [28], it is the archetypical W[1]-hard problem in parameterized complexity [14], and is one of the classic problems difficult to approximate [27].

In the light of the hardness of MWIS within multiple paradigms, one may ask what assumptions on the input make the problem easier. More formally, we can ask for which graph classes $\mathcal{G}$, the assumption that the input graph comes from $\mathcal{G}$ allows for faster algorithms for MWIS. For example, if $\mathcal{G}$ is the class of planar graphs, MWIS remains NP-hard, but the classic layering approach of Baker [6] yields a polynomial-time approximation scheme and simple kernelization arguments give a parameterized algorithm [13].

This motivates a more methodological study of the complexity of MWIS depending on the graph class $\mathcal{G}$ the input comes from. As the space of all graph classes is too wide and admits strange artificial examples, the arguably simplest regularization assumption is to restrict the attention to hereditary graph classes, i.e., graph classes closed under vertex deletion. Every hereditary graph class $\mathcal{G}$ can be characterized by *minimal forbidden induced subgraphs*: the (possibly infinite) set $\mathcal{F}$ of minimal (under vertex deletion) graphs that are not members of $\mathcal{G}$. Then, we have $G \in \mathcal{G}$ if and only if no member of $\mathcal{F}$ is an induced subgraph of G; when we want to emphasize the set $\mathcal{F}$, we refer to the graph class $\mathcal{G}$ as the class of *$\mathcal{F}$-free graphs* and shorten it to H-free graphs if $\mathcal{F} = \{H\}$.

If a problem turns out to be easier in a class of $\mathcal{F}$-free graphs, in many cases it is a single forbidden induced subgraph $H \in \mathcal{F}$ that is responsible for tractability, and the problem at hand is already easier in H-free graphs. A prime example of this phenomenon are the classes of line graphs and claw-free graphs. Recall that a *line graph* of a graph H is a graph G with $V(G) = E(H)$ where two vertices of G are adjacent if their corresponding edges in H are incident to the same vertex. Observe that MWIS in a line graph G of a graph H becomes the Maximum Weight Matching problem in the preimage graph H; a problem solvable in polynomial time by deep combinatorial techniques [15]. It turns out that the tractability of MWIS in line graphs can be explained solely by one of the minimal forbidden induced subgraphs for the class of line graphs, namely the *claw* $S_{1,1,1}$. (For integers $a, b, c \geqslant 1$, by $S_{a,b,c}$ we denote the tree with exactly three leaves, within distance a, b, and c from the unique vertex of degree 3.) As proven in 1980, MWIS is polynomial-time solvable already in the class of $S_{1,1,1}$-free graphs [39, 48], called also the class of *claw-free graphs* (for recent fast algorithms, see [16, 46]).

Together with the vastness of the space of all hereditary graph classes, this motivates us to focus on $\mathcal{F}$-free graphs for finite sets $\mathcal{F}$, in particular on the case $|\mathcal{F}| = 1$. This turned out to be particularly interesting for MWIS. As observed by Alekseev [3], for the "overwhelming majority" of finite sets $\mathcal{F}$, MWIS remains NP-hard on $\mathcal{F}$-free graphs. More precisely Alekseev observed that MWIS remains NP-hard on $\mathcal{F}$-free graphs unless, for at least one graph in $\mathcal{F}$, every connected component is a path or an $S_{a,b,c}$ for some integers a,b,c. Since the original NP-hardness proof of Alekseev [3] in 1982, no new finite sets $\mathcal{F}$ have been discovered such that MWIS remains NP-hard on $\mathcal{F}$-free graphs. We conjecture that this is because all of the remaining cases are actually solvable in polynomial time.

Conjecture 1.1. *For every H that is a forest whose every component has at most three leaves, Maximum Weight Independent Set is polynomial-time solvable when restricted to H-free graphs.*

To the best of our knowledge, the first place Conjecture 1.1 appeared explicitly is [31]. Let us remark that Conjecture 1.1, if true, would yield a dichotomy for the computational complexity of MWIS on $\mathcal{F}$-free graphs for all finite sets $\mathcal{F}$. Consider any $\mathcal{F}$ such that NP-hardness of MWIS on $\mathcal{F}$-free graphs does not follow from Alekseev's proof. It follows that the class of $\mathcal{F}$-free graphs is contained in the class of H-free graphs for some graph H for which polynomial time solvability of MWIS is conjectured in Conjecture 1.1.

From the positive side, as already mentioned, we know that MWIS is polynomial-time solvable in $S_{1,1,1}$-free graphs since 1980. Around the same time, it was shown that the class of P_4-free graphs (by P_t we denote the path on t vertices) coincides with the class of *cographs* and has very strong structural properties (in modern terms, has bounded cliquewidth) thus allowing efficient algorithms for MWIS and many other combinatorial problems. Over the years, we have witnessed a few scattered results for some special cases of H-free graphs, such as $S_{1,1,2}$-free graphs [4, 32], $2K_2$-free graphs [17], tK_2-free graphs [18], ℓP_3-free graphs [31], $\ell S_{1,1,1}$-free graphs [9], $tK_2 + P5$-free or $tK_2 + S_{1,1,2}$-free graphs [45], as well as progress limited to various subclasses (see [8, 8, 22, 26, 29, 33–37, 40–44] for older and newer results of this kind).

The research in the area got significant momentum in the last decade. The progress can be partitioned into two main threads. The first one focuses on the framework of *potential maximal cliques*, introduced by Bouchitté and Todinca [7], and focuses on providing polynomial-time algorithms for P_t-free graphs for small values of t. A landmark result here is due to Lokshtanov, Vatshelle, and Villanger [30] who were the first to show the usability of the framework in the context of P_t-free graphs by providing a polynomial-time algorithm for MWIS in P_5-free graphs. This has been later extended to P_6-free graphs [23] and related graph classes [2]. A notable property of this framework is that in most cases it not only provides algorithms for MWIS, but for a wide range of problems asking for large induced subgraph of small treewidth, for example Feedback Vertex Set.

The second thread attempts at treating P_t-free or $S_{t,t,t}$-free graphs in full generality, but relaxing the requirements on either the running time (by providing subexponential or quasi-polynomial-time algorithms) or the accuracy (by providing approximation algorithms, such as approximation schemes). Here, the starting point is the theorem of Gyárfás [24, 25] (see also [5]).

Theorem 1.2. *Every vertex-weighted graph G contains an induced path Q such that every connected component of $G - N[V(Q)]$ has weight at most half of the weight of G.*

As an induced path in a P_t-free graph has less than t vertices, a P_t-free graph admits a balanced separator (in the sense of Theorem 1.2) consisting of neighborhood of at most $t - 1$ vertices. In other words, P_t-free graphs admit a balanced separator dominated by $t - 1$ vertices. Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [10] observed that this easily gives a quasi-polynomial-time approximation scheme (QPTAS) for MWIS in P_t-free graphs, and they designed an elaborate argument involving the celebrated three-in-a-tree theorem of Chudnovsky and Seymour [12] to extend the result to the $S_{t,t,t}$-free case and H-free case where H is a forest of trees with at most three leaves each. Abrishami, Chudnovsky, Dibek, and Rzążewski[1] used also the three-in-a-tree theorem to obtain a polynomial-time algorithm for MWIS for $S_{t,t,t}$-free graphs of bounded degree. Gartland and Lokshtanov showed how to use the theorem of Gyárfás to design exact quasi-polynomial-time algorithm for MWIS in P_t-free graphs [19], for every fixed t. This algorithm was later simplified by Pilipczuk, Pilipczuk, and Rzążewski [47] and the union of the authors of these two papers showed that the approach works for a much wider class of problems and a slightly wider graph class [21]. Last year, Majewski, Masařík, Novotná, Okrasa, Pilipczuk, Rzążewski, and Sokołowski [38] gave a cleaner argument for an existence of a QPTAS for MWIS in $S_{t,t,t}$-free graphs.

This work provides the pinnacle of the second thread by showing that MWIS is quasi-polynomial-time solvable in all cases treated by Conjecture 1.1.

Theorem 1.3. *For every H that is a forest whose every component has at most three leaves, there is an algorithm for Maximum Weight Independent Set in H-free graphs running in time $n^{O_H(\log^{19} n)}$.*

Here O_H denotes constants depending on $|H|$ being repressed. Theorem 1.3 provides strong evidence in favor of Conjecture 1.1, as it refutes the existence of an NP-hardness proof for MWIS for H-free graphs as in Conjecture 1.1, unless all problems in NP can be solved in quasi-polynomial time.

2 OUR TECHNIQUES

As discussed in [19] (in particular Theorem 2), to show Theorem 1.3 it suffices to focus on the case $H = S_{t,t,t}$ for a fixed integer $t \geqslant 1$. Together with a simple self-reducibility argument, it is enough to prove the following.

Theorem 2.1. *For every integer $t \geqslant 1$, the maximum possible weight of an independent set in a given n-vertex $S_{t,t,t}$-free graph can be found in $n^{O_t(\log^{16}(n))}$ time.*

Here O_t denotes constants depending on t being repressed.

2.1 The Key Structural Result

While Theorem 1.2 provides a balanced separator consisting of a few neighborhoods in a P_t-free graph, it does not seem to be directly usable for $S_{t,t,t}$-free graphs. The example of G being a line graph of a clique (which is $S_{1,1,1}$-free) shows that we cannot hope for merely a balanced separator consisting of a few neighborhoods in $S_{1,1,1}$-free graphs.

However, if G is a line graph, MWIS is solvable in polynomial-time by a very different reason than Theorem 1.2: because it corresponds to a matching problem in the preimage graph. Luckily, there is a known formalism capturing decompositions of a graph that are "like a line graph": extended strip decompositions.

For a graph G, a *strip decomposition* consists of a graph H (called the *host*) and a function η that assigns to every edge $e \in E(H)$ a subset $\eta(e) \subseteq V(G)$ such that $\{\eta(e) \mid e \in E(H)\}$ is a partition of $V(G)$ and a subset $\eta(e,x) \subseteq \eta(e)$ for every endpoint $x \in e$ such that the following holds: for every $v_1, v_2 \in V(G)$ with $v_1 \in \eta(e_1)$, $v_2 \in \eta(e_2)$ and $e_1 \neq e_2$ we have $v_1v_2 \in E(G)$ if and only if there is a common endpoint $x \in e_1 \cap e_2$ with $v_1 \in \eta(e_1, x)$ and $v_2 \in \eta(e_2, x)$. Note that if G is the line graph of H, then G has a strip decomposition with host H and $\eta(e,x) = \eta(e,y) = \{e\}$ for every $xy = e \in E(H) = V(G)$. The crucial observation is that if one provides a strip decomposition (H,η) of a graph G together with, for every $xy \in E(H)$, the maximum possible weight of an independent set in $G[\eta(xy)]$, $G[\eta(xy) \setminus \eta(xy,x)]$, $G[\eta(xy) \setminus \eta(xy,y)]$, and $G[\eta(xy) \setminus (\eta(xy,x) \cup \eta(xy,y)]$ (these graphs are henceforth called *particles*), then we can reduce computing the maximum weight of an independent set in G to the maximum weight matching problem in the graph H with some gadgets attached [10].

An *extended strip decomposition* also allows vertex sets $\eta(x)$ for $x \in V(H)$ and triangle sets $\eta(xyz)$ for triangles xyz in H; a precise definition can be found in preliminaries, but is irrelevant for this overview. Importantly, the notion of a particle generalizes and the property that one can solve MWIS in G knowing the answers to MWIS in the particles is still true. Extended strip decompositions come from the celebrated solution to the *three-in-a-tree* problem by Chudnovsky and Seymour. The task is to determine if a graph contains an induced subgraph which is a tree connecting three given vertices. The following theorem says that The three-in-a-tree problem can be solved in polynomial time:

Theorem 2.2 ([12, Section 6], simplified version)**.** *Let G be an n-vertex graph and Z be a subset of vertices with $|Z| \geqslant 2$. There is an algorithm that runs in time $O(n^5)$ and returns one of the following:*

- *an induced subtree of G containing at least three elements of Z,*
- *an extended strip decomposition (H,η) of G where for every $z \in Z$ there exists a distinct degree-1 vertex $x_z \in V(H)$ with the unique incident edge $e_z \in E(H)$ and $\eta(e_z, x_z) = \{z\}$.*

In a sense, an extended strip decomposition as in Theorem 2.2 is a certificate that no three vertices of Z can be connected by an induced tree in G.

[10] combined Theorem 1.2 with Theorem 2.2 in a convoluted way to show a QPTAS for MWIS in $S_{t,t,t}$-free graphs; Thereom 2.2 is used here to construct an induced $S_{t,t,t}$ in the argumentation. [38] provided a simpler argument for the existence of a QPTAS: they derived from Theorem 2.2 the following structural result.

Theorem 2.3 ([38, Theorem 2] in a weighted setting)**.** *For every fixed integer t, there exists a polynomial-time algorithm that, given an n-vertex graph G with nonnegative vertex weights, either:*

- *outputs an induced copy of $S_{t,t,t}$ in G, or*
- *outputs a set $\mathcal{P}$ consisting of at most $11 \log n + 6$ induced paths in G, each of length at most $t+1$, and a rigid extended strip decomposition of $G - N[\bigcup \mathcal{P}]$ with every particle of weight at most half of the total weight of $V(G)$.*

(Here, rigid means that the extended strip decomposition does not have some unnecessary empty sets; in a rigid decomposition the size of H is bounded linearly in the size of G. The formal statement of Theorem 2.3 in [38] is only for uniform weights in G, but as observed in the conclusions of [38], the proof works for arbitrary vertex weights.)

[38] showed that Theorem 2.3 easily gives a QPTAS for MWIS in $S_{t,t,t}$-free graphs, along the same lines as how [10] showed that Theorem 1.2 easily gives a QPTAS for MWIS in P_t-free graphs.

However, it seems that the outcome of Theorem 2.3 is not very useful if one aims for an exact algorithm faster than a subexponential one. Our main graph-theoretic contribution is a strengthening of Theorem 2.3 to the following.

Theorem 2.4. *For every fixed integer t, there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G and a weight function $\mathfrak{w} : V(G) \to [0,+\infty)$, returns one of the following outcomes:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a subset $X \subseteq V(G)$ of size at most $c_t \cdot \log(n)$ such that every component of $G - N[X]$ has weight at most $0.99\mathfrak{w}(G)$;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than $0.5\mathfrak{w}(G)$.*

That is, we either provide an extended strip decomposition of the *whole* graph (not only after deleting a neighborhood of a small number of vertices as in Theorem 2.3) or a small number of vertices such that deletion of their neighborhood breaks the graph into multiplicatively smaller (in terms of weight) components.

The proof of Theorem 2.4 is provided in the full version of paper [20, Section 3]. Let us briefly sketch it. We start by applying Theorem 2.3 to G; we are either already done or we have a set $Z := \bigcup_{P\in\mathcal{P}} V(P)$ of size $O(\log n)$ and an extended strip decomposition (H,η) of $G - N[Z]$ with small particles. Our goal is now to add the vertices of $N[Z]$ one by one back to (H,η), possibly exhibiting one of the other outcomes of Theorem 2.4 along the way. That is, we want to prove the following lemma:

Lemma 2.5. *For every fixed integer t there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G, a weight function $\mathfrak{w} : V(G) \to [0,+\infty)$, a real $\tau \geqslant \mathfrak{w}(G)$, a vertex $v \in V(G)$, and a rigid extended strip decomposition (H,η) of $G - v$ with every particle of weight at most 0.5τ, returns one of the following:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a set $Z \subseteq V(G)$ of size at most c_t such that every connected component of $G - N[Z]$ has weight at most 0.99τ;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than 0.5τ.*

A simple yet important observation for Lemma 2.5 is that for $x \in V(H)$ of degree at least two, the set $\bigcup_{y\in N_H(x)} \eta(xy,x)$ can be dominated by at most two vertices, as the sets $\eta(xy,x)$ for $y \in N_H(x)$ are complete to each other. Consequently, if (A,B) is a separation in H of small order, then the part of G that is placed by η in $H[A]$ and the part of G that is placed by η in $H[B]$ can be separated by deleting at most $2|A\cap B|$ vertex neighborhoods in G. Hence, if there is a separation (A,B) in H of constant order where both sides of this separation have substantial weight (at least 0.01τ), we can provide the second outcome of Lemma 2.5.

As $N[v]$ is just one neighborhood, the same observation holds if, instead of looking at (H,η), we look at the inherited extended strip decomposition (H',η') of $G - N[v]$. Here, (H',η') is obtained from (H,η) by first deleting vertices of $N(v)$ from sets $\eta(\cdot)$ and then performing a cleanup operation that trims unnecessary empty sets and ensures that for every $xy \in E(H')$ there is a path in $G[\eta'(xy)]$ between $\eta'(xy,x)$ and $\eta'(xy,y)$. Hence, we can take all separations (A,B) in H' of order bounded by a large constant (depending on t) and orient them from the side that contains less than 0.01τ weight to the side containing almost all the weight of G. This orientation defines a tangle in H'. By classic results from the theory of graph minors, this tangle implies the existence of a large wall W in H' which is always mostly on the "large weight" side of any separation (A,B) of constant order. The cleaning operation ensures that the wall W is also present in (H,η).

An important observation now is that, because (H',η') is cleaned as described below, any family of vertex-disjoint paths in H' projects down to a family of induced, vertex-disjoint, and anti-adjacent paths in G of roughly the same length (or longer): for a path P in H, just follow paths from $\eta(xy,x)$ to $\eta(xy,y)$ in $G[\eta(xy)]$ for consecutive edges xy on P. Furthermore, a wall W is an excellent and robust source of long vertex-disjoint paths.

This allows us to prove that if the neighbors of v are well-connected to the wall W in (H,η) — either they are spread around the wall itself, or one can connect them to W via three vertex-disjoint paths in H — then G contains an induced $S_{t,t,t}$. Otherwise, we show that there is a separation (A,B) in H with the neighbors of v essentially all contained in the sets of $H[A]$, while W lies on

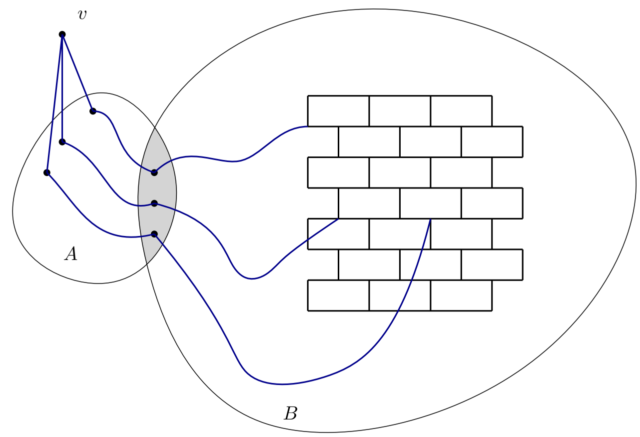

Figure 1: Extending a subdivided claw in G_A to an $S_{t,t,t}$ using the large wall W in B.

the B-side of the separation. (Here, a large number of technical details are hidden in the phrase "essentially contained".) We construct a graph G_A being the subgraph of G induced by the vertices contained in the η sets of $H[A]$, augmented with a set Z of artificial vertices attached to $\bigcup_{y\in N_H(x)\cap A} \eta(xy,x)$ for $x \in A\cap B$; vertices of Z signify possible "escape paths" to the wall W. These "escape paths" allow us to show that any induced tree in G_A that contains at least three vertices of Z lifts to an induced $S_{t,t,t}$ in G, see Figure 1. Hence, the algorithm of Theorem 2.2 applied to G_A and Z can be used to rebuild $H[A]$ to accommodate v there as well, or to expose an induced $S_{t,t,t}$. This finishes the sketch of the proof of Lemma 2.5 and of Theorem 2.4.

We would like to highlight a significant difference between previous works [1, 10, 38] and our use of the three-in-a-tree theorem to exhibit an $S_{t,t,t}$ in a graph or obtain an extended strip decomposition. All aforementioned previous works essentially picked three anti-adjacent paths P_1, P_2, P_3 of length t each, with endpoints say x_i and y_i for $i = 1,2,3$, removed their neighborhood except for the neighbors of y_is, and called three-in-a-tree for the set $Z = \{x_1,x_2,x_3\}$; note that any induced tree in the obtained graph that contains Z contains also an induced $S_{t,t,t}$. This method inherently produced extended strip decompositions not for the entire graph, but only for after removal of a number of neighborhoods. Furthermore, it used the assumption of being $S_{t,t,t}$-free only in a very local sense: there is no $S_{t,t,t}$ with paths extendable to the given three vertices of Z. In this work, in contrast, we apply the three-in-a-tree theorem to a potentially much bigger set Z, and use a subdivided wall in the host graph of the extended strip decomposition to extend any induced tree found to an induced $S_{t,t,t}$. In this way, we used the assumption of being $S_{t,t,t}$-free in a more global way than just merely asking for three particular leaves.

2.2 Branching

We now proceed with a sketch of our recursive branching algorithm. On a very high level, it is based on techniques used in the quasi-polynomial time algorithm for independent set on P_k-free graphs found in [19], though multiple new ideas are required to make the reasoning work in the setting of $S_{t,t,t}$-free graphs, making both the algorithm and its running time analysis quite a bit more technical. We will soon sketch the algorithm found in [19] and describe how to extend it to $S_{t,t,t}$-free graphs, but first we must address a major barrier. The fact that P_k-free graphs have balanced separators dominated by k vertices, as discussed after Theorem 1.2, is a crucial fact used in the algorithm of [19]. But, as mentioned previously, $S_{t,t,t}$-free graphs have no such property (take for instance the line graph of a clique). This is where Theorem 2.4 comes to the rescue.

When applying Theorem 2.4 to G (the input graph of the current call of the algorithm), since we assume that G is $S_{t,t,t}$-free, we are guaranteed that outcome (1) will not occur. If outcome (3) occurs then we get an extended strip decomposition (H, η) and, as previously mentioned, we can reduce finding a maximum independent set of G to finding a maximum independent set in each particle of (H, η). That is great news, as each particle has at most half of the weight of G, and we can easily employ a divide-and-conquer strategy by recursively calling the algorithm on each particle of (H, η). So, since outcome (1) never happens and outcome (3) gives us an easy algorithm, we can always assume that outcome (2) happens, that is, that Theorem 2.4 gives us a balanced separator of G that is dominated by $O(\log n)$ vertices, and now we can try to extend the techniques found in [19] to work for $S_{t,t,t}$-free graphs. Therefore, for the rest of this subsection we will focus on sketching an algorithm for independent set on an $S_{t,t,t}$-free graph G such that all induced subgraphs of G have a balanced separator dominated by some constant number of vertices (the stronger assumption of a constant number of vertices versus $\log n$ vertices does not change the algorithm very much and simplifies the discussion).

Before sketching the algorithm let us give a few short definitions around balanced separators for an $S_{t,t,t}$-free graph G. For $n' > 0$, we say that a set $S \subseteq V(G)$ is a *n'-balanced separator* for G if no component of $G - S$ has more than n' vertices. If $A \subseteq V(G)$ and no component of $G - S$ contains over n' vertices of A, we say that S is a n'-balanced separator for (G, A). The outcome (2) of Theorem 2.4 gives us a $0.99|A|$-balanced separator for (G, A) dominated by $O(\log n)$ vertices (again here for simplicity we will assume that these balanced separators are in fact dominated by a constant number of vertices). However, by picking a constant number of balanced separators as provided by Theorem 2.4 and taking their union, we can obtain $c|A|$-balanced separators for (G, A) dominated by a constant number of vertices for any fixed $c \in (0, 1)$, so we will assume we have access to such strengthened balanced separators.

Summary of the Quasi-Polynomial Time Algorithm for MWIS on P_k-free Graphs. The starting point for our algorithm is the algorithm for MWIS on P_k-free graphs by Gartland and Lokshtanov [19], who in turn build on an algorithm of Bacsó, Lokshtanov, Marx, Pilipczuk, Tuza, and van Leeuwen [5]. We therefore give a brief summary of these algorithms.

We first consider the simple $n^{O(k\sqrt{n}\log n)}$ time algorithm of [5] for MWIS on P_k-free graphs. We begin with an n-vertex P_k-free graph G and branch on all vertices of degree at least $\sqrt{n}$: we either exclude such a vertex from the solution (and thus remove it from the graph), or we include it (and then remove its whole neighborhood from the graph). After this we may assume that the graph in our current instance (we will still refer to this graph as G although some vertices of the original graph G have been removed) now has maximum degree at most $\sqrt{n}$. We solve this instance by finding an $n/2$-balanced separator, S, for G that is dominated by at most k vertices. Since G has maximum degree $\sqrt{n}$ and S is dominated by at most k vertices, S can have size at most $k\sqrt{n}$. We then branch on all $k\sqrt{n}$ vertices of S simultaneously, which then breaks up the graph into small connected components and we recurse on each component. A simple analysis shows that this runs in $n^{O(k\sqrt{n}\log n)}$ time.

Now, let us try to improve it to an algorithm that runs in time $n^{O(kn^{1/3}\log n)}$. We first state a modified form of a lemma that appears in [19].

Lemma 2.6. *Let G be an n-vertex P_k-free graph and $\mathcal{F}$ a multi-set of subsets of $V(G)$ such that for every $S \in \mathcal{F}$ no component of $G - S$ has more than $n/2$ vertices. Assume that no vertex belongs to more than c sets of $\mathcal{F}$ counting multiplicity. Then provided $|\mathcal{F}| \geqslant 3ck$, no component of G contains more than $3n/4$ vertices.*

Sketch of proof. Let $S \in \mathcal{F}$ and assume for a contradiction that the largest component of $G - S$, call it C, has more than $3n/4$ vertices. Select vertices a, b uniformly at random from C. As $|C| > 3n/4$ the probability that a and b belong to different components of $G - S$ is at least $1/3$. If we let X_S be the random variable that is 1 if a and b are in different components of $G - S$ and 0 otherwise, then $\mathbb{E}[X_S] \geqslant \frac{1}{3}$. By the linearity of expectation, we have $\mathbb{E}[\sum_{S \in \mathcal{F}} X_S] \geqslant \frac{1}{3} \cdot 3ck \geqslant ck$. It follows that there exists vertices $a, b \in S$ such that for at least ck sets, S', in $\mathcal{F}$ (counting multiplicity) a and b are in different components of $G - S'$. Let $\mathcal{F}'$ be the subset of $\mathcal{F}$ that contains these sets S'. It follows that for any induced path P with a and b as its endpoints, if $S' \in \mathcal{F}'$ then $V(P) \cap S' \neq \emptyset$. Since $\mathcal{F}'$ has at least ck sets and no vertex of P belongs to more than c sets in $\mathcal{F}'$, P must have at least k vertices, contradicting the assumption that G is P_k-free. ◻

For the $n^{O(kn^{1/3}\log n)}$ algorithm, we again begin by branching on vertices of high degree, but this time we set the threshold to vertices with degree at least $n^{2/3}$. After this we may assume the graph in our current instance, call it G^1, has maximum degree $n^{2/3}$. We then find a balanced separator, S^1, for G^1 that is dominated by k vertices, hence S^1 has at most $kn^{2/3}$ vertices. We then branch on all vertices with at least $n^{1/3}$ neighbors in S^1. Now we assume the graph considered in our current instance, call it G^2, has maximum degree $n^{2/3}$ and a balanced separator S^1 such that no vertex of G^2 has more than $n^{1/3}$ neighbors in S^1. We then find a balanced separator, S^2, for G^2 that is dominated by k vertices, hence S^2 has at most $kn^{2/3}$ vertices and $S^1 \cap S^2$ has size at most $kn^{1/3}$. We then branch on all vertices with at least $n^{1/3}$ vertices in S^2 and we branch on all vertices that belong to $S^1 \cap S^2$, so S^1 and S^2 "become disjoint". We repeat this $3k$ times until we are in an instance where we have a graph G^{3k} and $3k$ pairwise disjoint balanced separators $S^1, \ldots S^{3k}$. By Lemma 2.6, G^{3k} has no component with over $3n/4$ vertices and we then recurse on each component. A somewhat more involved, but still fairly simple analysis shows that this runs in $n^{O(kn^{1/3}\log n)}$ time.

In the $n^{O(kn^{1/3}\log n)}$-time algorithm, we branched on vertices that: had over $n^{2/3}$ neighbors, or had $n^{1/3}$ neighbors in any of the balanced separators we picked up, or belonged to two of the balanced separators we picked up. In order to modify this algorithm to run in quasi-polynomial time all that must be done is change the branching threshold. In particular, the algorithm collects balanced separators (each dominated by at most k vertices) and will branch on any vertex that has over $n/2^i$ neighbors that belong to i or more of the collected balanced separators (the algorithm no longer branches on vertices that only have high degree). Any vertex that belongs to $\log n$ of the collected balanced separators will then be branched on, so no vertex will ever belong to more than $\log n$ of the collected balanced separators. So, by Lemma 2.6, after collecting $3k\log n$ of these balanced separators, the graph will not have any large component. A runtime analysis of this algorithm shows that it runs in quasi-polynomial time. Note that in all three algorithms discussed here (the $n^{O(kn^{1/2}\log n)}$-time, $n^{O(kn^{1/3}\log n)}$-time, and quasi-polynomial-time algorithm) it is crucial for efficient runtime that the balanced separators we use are dominated by few vertices (they were dominated by k vertices here, but being dominated by polylog(n) vertices would still be sufficent).

Back to $S_{t,t,t}$-free Graphs. Recall that we wish to get a quasi-polynomial time algorithm for MWIS on $S_{t,t,t}$-free graphs for the case where every induced subgraph of the input graph G has a set S of at most c_t vertices such that $N[S]$ is a $n/2$-balanced separator. Up to the bound on the set dominating the separator, this is precisely the case when we keep getting outcome (2) whenever we apply Theorem 2.4.

We want to mimic the algorithm for P_k-free graphs. This algorithm used that the input graph is P_k-free in precisely two places. The first is to keep getting constant size sets S such that $N[S]$ is an $n/2$-balanced separator. This is easily adapted to our new setting because we keep getting such sets whenever we apply Theorem 2.4.

The second place where P_k-freeness is used is in Lemma 2.6, which states that a P_k-free graph cannot have a set of $3k\log n$ balanced separators such that no vertex of G appears in at most $O(\log n)$ of them. If we could strengthen the statement of Lemma 2.6 to $S_{t,t,t}$-free graphs we would be done! Unfortunately, such a strengthening is false, indeed a path is a counterexample (each vertex close to the middle of the path is a balanced separator).

Nevertheless, a subtle weakening of Lemma 2.6 does turn out to be true. In particular, in $S_{t,t,t}$-free graphs it is not possible to pack "very strong" balanced separators that are dominated by "very few" vertices. We will call such balanced separators *c-boosted balanced separators.* A somewhat simplified definition of a c-boosted balanced separator is a set $N[S]$ dominated by a set S of at most c vertices, such that no component of $G-N[S]$ has more than $|V(G)|/16c^2$ vertices. It turns out that on $S_{t,t,t}$-free graphs Lemma 2.6 is true if "balanced separators" are replaced by "s-boosted balanced separators" for appropriately chosen integer s.

Lemma 2.7. *Let G be an n-vertex $S_{t,t,t}$-free graph, s an integer, and $\mathcal{F}$ a multi-set of subsets of $V(G)$ such that every set in $\mathcal{F}$ is an s-boosted balanced separator. Assume no vertex belongs to more than c sets of $\mathcal{F}$. Then, provided $|\mathcal{F}| \geqslant 80sct$, no component of G contains over $3n/4$ vertices.*

We skip sketching the proof of Lemma 2.7 here (see the full version [20, Section 4.2.2] for a formal statement and proof of this lemma), but we will remark that one of the key ingredients of the proof is a probabilistic argument akin to the proof of Lemma 2.6.

At this point we are one "disconnect" away from being able to utilize the strategy for P_k free graphs: Theorem 2.4 keeps giving us balanced separators, while Lemma 2.7 tells us that we can't pack *boosted* balanced separators. Indeed, if we assumed our $S_{t,t,t}$-free graphs always had, say, c_t-boosted balanced separators (where c_t is some constant that depends on t), then by the exact same reasoning as before, the strategy of iteratively collecting a c_t-boosted balanced separator and then branching (on all vertices that have over $n/2^i$ neighbors that belong to i or more of the collected c_t-boosted balanced separators) would work. Any vertex that belongs to $\log n$ of the collected c_t-boosted balanced separators will then be branched on, so no vertex will ever belong to over $\log n$ of the collected balanced separators. So, by Lemma 2.7, after collecting $80c_t t\log n$ of these c_t-boosted balanced separators, the graph will not have any large component. A running time analysis identical to the one for P_k-free graphs [19] would then show that this algorithm runs in quasi-polynomial time.

Is it possible to bridge the "disconnect" from the other side and keep getting *boosted* balanced separators? This looks difficult, but we are able to bridge the gap algorithmically, by branching in such a way that a "normal" balanced separator becomes boosted. We can then add this boosted balanced separator to our collection of previously created boosted balanced separators, and then apply Lemma 2.7 to this collection to conclude that the graph gets sufficiently disconnected before the collection grows too large. We now sketch how to "boost" a separator.

Boosting Separators. We begin with a balanced separator $N[S]$, dominated by a set S of at most c_t vertices, such that no component of $G-N[S]$ has more than $n/2$ vertices. (For technical reasons in the actual algorithm $N[S]$ is not a balanced separator, but rather a set given by Theorem 2.3 so that $G-N[S]$ has an extended strip decomposition with no large particles; from the viewpoint of efficient independent set algorithms this is just as useful.) We wish to turn $N[S]$ into a c_t-boosted balanced separator. In order to do this, we consider all vertices of $N[S]$ that have a neighbor in a large component of $G-N[S]$; we call this set relevant(G,S) (see Figure 2. This is a slight simplification of the actual definition of relevant(G,S) that we use in the algorithm). By "large component" we mean any component of $G-N[S]$ that has more than $n/16c_t^2$ vertices (note that if there are no such components, then $N[S]$ is a c_t-boosted balanced separator). In order to branch in a way that turns $N[S]$ into a c_t-boosted balanced separator, we use the following lemma, similar to Lemmas 2.6 and 2.7.

Lemma 2.8. *Let G be an n-vertex $S_{t,t,t}$-free graph, let $N[S]$ be a balanced separator for G dominated by a set S of at most c_t vertices, and let $\mathcal{F}$ be a multi-set of $|relevant(G,S)|/100c_t^3$-balanced separators for $(G, relevant(G,S))$. Assume no vertex belongs to over c sets of $\mathcal{F}$. If $|\mathcal{F}| \geqslant 10ct$, either S is a c_t-boosted balanced separator or no component of G contains more than $3n/4$ vertices.*

The proof of Lemma 2.8 follows a similar "expectation argument" that Lemma 2.6 uses, although it is a bit more involved. We do not

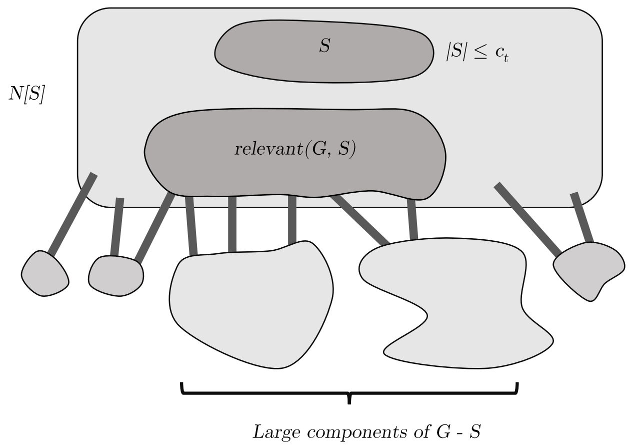

Figure 2: Illustration of how the set relevant(G, S) is obtained from S.

sketch a proof of Lemma 2.8 here (this lemma statement is more or less a combination of Observation 4.6 and Lemma 4.9 given in the full version [20]).

This lemma suggests the following branching strategy. We first pick up an $n/2$-balanced separator $N[S]$ dominated by a set S of c_t vertices, and we will try use Lemma 2.8 to turn $N[S]$ into a c_t-boosted balanced separator or break up G into small components. We use the same reasoning as before: iteratively collect $|\mathsf{relevant}(G,S)|/100c_t^3$-balanced separators for $(G, \mathsf{relevant}(G,S))$ and branch (on all vertices that have over $n/2^i$ neighbors that belong to i or more of the collected balanced separators). Any vertex that belongs to $\log n$ of the collected balanced separators will then be branched on, so no vertex will ever belong to over $\log(n)$ of the collected balanced separators. So, by Lemma 2.8 after collecting $10t \log n$ of these $|\mathsf{relevant}(G,S)|/100c_t^3$-balanced separators for $(G, \mathsf{relevant}(G,S))$, either the graph will have no large component (and then we make large progress by calling the algorithm recursively on the components) or S is now a c_t-boosted balanced separator, which we then add to our collection of c_t-boosted balanced separators. By Lemma 2.7 this collection cannot grow larger than $80c_t t \log n$ before our graph no longer has large connected components.

The running time analysis of this algorithm essentially looks like this: if we could assume that boosting a single balanced separator to become a boosted balanced separator took constant time, then the analysis would be more or less identical to the analysis of the algorithm for MWIS on P_k-free graphs. However, now each individual "boosting" step is instead a branching algorithm whose analysis again is very similar to the analysis of the algorithm for MWIS on P_k-free graphs, so each boosting step corresponds to a recursive algorithm with quasi-polynomially many leaves. Since quasi-polynomial functions compose the entire running time is still quasi-polynomial. Finally we need to take into account what would happen if outcome (3) of Theorem 2.4 does occur, but this can fairly easily be shown to only be good for the progress of the algorithm.

3 CONCLUSION

Let us point out some possible directions for future research. First, on the structural side, we believe that Theorem 2.4 could be improved so that in the second outcome the balanced separator is dominated by a constant (depending on t) number of vertices. The only reason why the current statement has the logarithmic bound is that in Theorem 2.3 the number of deleted neighborhoods is logarithmic. [38] conjectured that Theorem 2.3 can actually be improved so that the number of deleted neighborhoods is constant. Proving this conjecture would immediately yield an improved version of our Theorem 2.4. However, such a stronger version, while being more elegant, would not give any essentially new algorithmic result: the running time of our algorithms would still be quasi-polynomial (though a bit faster).

On the algorithmic side, an obvious natural problem is to provide a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs, for all t. While we believe that extended strip decompositions are the right tool to use towards this goal, it seems that decompositions like the ones obtained by Theorem 2.4 would not lead to such a statement. This is because recursing into a polynomial number of multiplicatively smaller particles inherently leads to a quasi-polynomial running time. We believe the ultimate goal would be to build an extended strip decomposition where each particle induces a graph from some "simple" class. In particular, so that we can solve MWIS for each particle in polynomial time without using recursion. Such decompositions for the simplest case, i.e., claw-free graphs, are provided by a deep structural result of Chudnovsky and Seymour [11].

An important milestone on the way towards obtaining a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs is to solve the case of P_t-free graphs, which is already a very ambitious goal.

ACKNOWLEDGMENTS

We would like to thank Amálka Masaříková and Jana Masaříková for useful discussions while working on this paper. Additionally, 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.

Pe. G. and Da. L. were supported by NSF award CCF-2008838. This work is a part of the project BOBR (Mi. P. and P. Rz.) 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). To. M. was supportesectionsh National Science Centre SONATA-17 grant number 2021/43/D/ST6/03312. Ma. P. was supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143. Ma. P. is also part of BARC, supported by the VILLUM Foundation grant 16582.

REFERENCES

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

[2] Tara Abrishami, Maria Chudnovsky, Marcin Pilipczuk, Paweł Rzążewski, and Paul D. Seymour. 2021. Induced subgraphs of bounded treewidth and the container method. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021*, Dániel Marx (Ed.). SIAM, 1948–1964. https://doi.org/10.1137/1.9781611976465.116

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

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

[5] Gábor Bacsó, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Zsolt Tuza, and Erik Jan van Leeuwen. 2019. Subexponential-Time Algorithms for Maximum Independent Set in P_t-Free and Broom-Free Graphs. *Algorithmica* 81, 2 (2019), 421–438. https://doi.org/10.1007/s00453-018-0479-5

[6] Brenda S. Baker. 1994. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *J. ACM* 41, 1 (1994), 153–180. https://doi.org/10.1145/174644.174650

[7] Vincent Bouchitté and Ioan Todinca. 2001. Treewidth and Minimum Fill-in: Grouping the Minimal Separators. *SIAM J. Comput.* 31, 1 (2001), 212–232. https://doi.org/10.1137/S0097539799359683

[8] Andreas Brandstädt and Raffaele Mosca. 2018. Maximum Weight Independent Sets for (P_7, triangle)-free graphs in polynomial time. *Discret. Appl. Math.* 236 (2018), 57–65. https://doi.org/10.1016/j.dam.2017.10.003

[9] Andreas Brandstädt and Raffaele Mosca. 2018. Maximum weight independent set for ℓclaw-free graphs in polynomial time. *Discrete Applied Mathematics* 237 (mar 2018), 57–64. https://doi.org/10.1016/j.dam.2017.11.029

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

[11] Maria Chudnovsky and Paul D. Seymour. 2008. Claw-free graphs. V. Global structure. *J. Comb. Theory, Ser. B* 98, 6 (2008), 1373–1410. https://doi.org/10.1016/j.jctb.2008.03.002

[12] Maria Chudnovsky and Paul D. Seymour. 2010. The three-in-a-tree problem. *Comb.* 30, 4 (2010), 387–417. https://doi.org/10.1007/s00493-010-2334-4

[13] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized algorithms.* Springer. https://doi.org/10.1007/978-3-319-21275-3

[14] Rod G. Downey and Michael R. Fellows. 1995. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science* 141, 1–2 (April 1995), 109–131. https://doi.org/10.1016/0304-3975(94)00097-3

[15] Jack Edmonds. 1965. Paths, Trees, and Flowers. *Canadian Journal of Mathematics* 17 (1965), 449–467. https://doi.org/10.4153/CJM-1965-045-4

[16] Yuri Faenza, Gianpaolo Oriolo, and Gautier Stauffer. 2011. An Algorithmic Decomposition of Claw-Free Graphs Leading to an $O(n^3)$-Algorithm for the Weighted Stable Set Problem. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)* (San Francisco, California). Society for Industrial and Applied Mathematics, USA, 630–646. https://doi.org/10.1137/1.9781611973082.49

[17] Martin Farber. 1989. On diameters and radii of bridged graphs. *Discrete Math.* 73, 3 (1989), 249–260. https://doi.org/10.1016/0012-365X(89)90268-9

[18] Martin Farber, Mihály Hujter, and Zsolt Tuza. 1993. An upper bound on the number of cliques in a graph. *Networks* 23, 3 (May 1993), 207–210. https://doi.org/10.1002/net.3230230308

[19] Peter Gartland and Daniel Lokshtanov. 2020. Independent Set on P_k-Free Graphs in Quasi-Polynomial Time. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16–19, 2020*, Sandy Irani (Ed.). IEEE, 613–624. https://doi.org/10.1109/FOCS46700.2020.00063

[20] Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. 2023. Maximum Weight Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time. arXiv:2305.15738 [cs.DS]

[21] Peter Gartland, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. 2021. Finding large induced sparse subgraphs in $C_{>t}$-free graphs in quasipolynomial time. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, Samir Khuller and Virginia Vassilevska Williams (Eds.). ACM, 330–341. https://doi.org/10.1145/3406325.3451034

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

[23] Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michał Pilipczuk. 2022. Polynomial-time Algorithm for Maximum Weight Independent Set on P_6-free Graphs. *ACM Trans. Algorithms* 18, 1 (2022), 4:1–4:57. https://doi.org/10.1145/3414473

[24] András Gyárfás. 1975. On Ramsey covering-numbers. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II.* Number 10 in Colloq. Math. Soc. Janos Bolyai. North-Holland, Amsterdam, 801–816.

[25] András Gyárfás. 1987. Problems from the world surrounding perfect graphs. In *Proceedings of the International Conference on Combinatorial Analysis and its Applications, (Pokrzywna, 1985) (Zastos. Mat., 19)*. 413–441. https://doi.org/10.4064/am-19-3-4-413-441

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

[27] Johan Håstad. 1999. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Math.* 182, 1 (1999), 105–142. https://doi.org/10.1007/BF02392825

[28] Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA (The IBM Research Symposia Series)*, Raymond E. Miller and James W. Thatcher (Eds.). Plenum Press, New York, 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9

[29] Ngoc Chi Lê, Christoph Brause, and Ingo Schiermeyer. 2015. The Maximum Independent Set Problem in Subclasses of $S_{i,j,k}$-Free Graphs. *Electron. Notes Discret. Math.* 49 (2015), 43–49. https://doi.org/10.1016/j.endm.2015.06.008

[30] Daniel Lokshtanov, Martin Vatshelle, and Yngve Villanger. 2014. Independent Set in P_5-Free Graphs in Polynomial Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014*, Chandra Chekuri (Ed.). SIAM, 570–581. https://doi.org/10.1137/1.9781611973402.43

[31] Vadim Lozin. 2017. From matchings to independent sets. *Discrete Appl. Math.* 231 (Nov. 2017), 4–14. https://doi.org/10.1016/j.dam.2016.04.012

[32] Vadim V. Lozin and Martin Milanič. 2008. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *J. Discrete Algorithms* 6, 4 (2008), 595–604. https://doi.org/10.1016/j.jda.2008.04.001

[33] Vadim V. Lozin, Martin Milanič, and Christopher Purcell. 2014. Graphs Without Large Apples and the Maximum Weight Independent Set Problem. *Graphs Comb.* 30, 2 (2014), 395–410. https://doi.org/10.1007/s00373-012-1263-y

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

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

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

[37] Frédéric Maffray and Lucas Pastor. 2016. Maximum Weight Stable Set in (P_7, bull)-free graphs. *CoRR* abs/1611.09663 (2016). http://arxiv.org/abs/1611.09663

[38] Konrad Majewski, Tomáš Masařík, Jana Masaříková, Karolina Okrasa, Marcin Pilipczuk, Paweł Rzążewski, and Marek Sokołowski. 2024. Max Weight Independent Set in Graphs with No Long Claws: An Analog of the Gyárfás' Path Argument. *ACM Transactions on Computation Theory* 16, 2 (March 2024), 8:1–8:18. https://doi.org/10.1145/3636422

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

[40] Raffaele Mosca. 1999. Stable Sets in Certain P_6-free Graphs. *Discret. Appl. Math.* 92, 2-3 (1999), 177–191. https://doi.org/10.1016/S0166-218X(99)00046-3

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

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

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

[44] Raffaele Mosca. 2021. Independent Sets in (P_{4+4}, Triangle)-Free Graphs. *Graphs Comb.* 37, 6 (2021), 2173–2189. https://doi.org/10.1007/s00373-021-02340-7

[45] Raffaele Mosca. 2022. New results on independent sets in extensions of $2K_2$-free graphs. *Graphs Comb.* 38, 4 (Aug. 2022). https://doi.org/10.1007/s00373-022-02532-9

[46] Paolo Nobili and Antonio Sassano. 2021. An $O(n^2 \log(n))$ algorithm for the weighted stable set problem in claw-free graphs. *Mathematical Programming* 186 (2021), 409–437. https://doi.org/10.1007/s10107-019-01461-5

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

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