Independent Set in P_5-Free Graphs in Polynomial Time

Daniel Lokshantov* Martin Vatshelle* Yngve Villanger*

October 9, 2013

Abstract

The Independent Set problem is NP-hard in general, however polynomial time algorithms exist for the problem on various specific graph classes. Over the last couple of decades there has been a long sequence of papers exploring the boundary between the NP-hard and polynomial time solvable cases. In particular the complexity of Independent Set on P_5-free graphs has received significant attention, and there has been a long list of results showing that the problem becomes polynomial time solvable on sub-classes of P_5-free graphs. In this paper we give the first polynomial time algorithm for Independent Set on P_5-free graphs. Our algorithm also works for the Weighted Independent Set problem.

1 Introduction

An independent set (also known as stable set) in a graph G is a set of pairwise non-adjacent vertices. In the Independent Set problem we are given as input a graph G on n vertices and the task is to find the largest independent set in G. In the weighted variant of the problem each vertex v comes with a non-negative weight w_v and the goal is to find the independent set I in G that maximizes $\sum_{v \in I} w_v$. The problem has numerous applications, including train dispatching [22] and data mining [48]. Independent Set is NP-complete, in fact it is one of the 21 problems proved NP-complete by Karp [33] in 1972. On general graphs Independent Set is hard to approximate within a factor of $O(n^\epsilon)$ for $\epsilon < 1$ [4], is not fixed parameter tractable unless FPT=W[1] [21], and admits no subexponential time algorithm under the Exponential Time Hypothesis [32, 36]. This motivates the study of Independent Set on restricted graph classes, and there is a wealth of research on the complexity of Independent Set on graphs with a structural constraint. For an example the problem becomes polynomial time solvable on bipartite graphs [17], but remains NP-complete on planar graphs. However, on planar graphs Independent Set admits polynomial time approximation schemes [35, 5] as well as subexponential time exact and parameterized algorithms [35].

Considerable effort has gone into classifying the graph classes for which the Independent Set problem becomes polynomial time solvable, and for which it remains NP-complete. A seminal result of Grötschel et al. [28] shows that Independent Set can be solved in polynomial time on perfect graphs. Polynomial time algorithms have also been found for $k \times K_2$-free graphs [6], graphs of bounded clique-width [19], and many others (see [13]). On the other hand the problem remains NP-complete even on planar graphs of maximum degree 3 [24], unit disc graphs [16], triangle-free graphs [44] and AT-free graphs [14]. While a complete classification of the complexity status of Independent Set on all graph classes is out of reach, achieving such a classification for all classes of graphs excluding a single graph H as an induced subgraph (we call such graphs H-free) seems like a more feasible goal, in particular if H is connected. It was noted by Alekseev [1] that Independent Set remains NP-complete on H-free graphs whenever H is connected, but neither a path nor a subdivision of the claw ($K_{1,3}$).

Graphs excluding the path P_3 are disjoint unions of cliques and so Independent Set is trivially solvable in polynomial time on P_3-free graphs. The graphs excluding a path P_4 on four vertices are known as cographs, and a linear time algorithm for Independent Set on cographs was shown by Corneil et al. [18] in 1981. For claw-free graphs, Sbihi [47] and Minty [40] showed polynomial time algorithms in 1980. There are two graphs H on five vertices for which NP-completeness of Independent Set on H-free graphs does not follow from Alekseev's result, namely the path P_5 on five vertices and the fork, which is obtained from the claw $K_{1,3}$ by subdividing any one of its edges. For fork-free graphs the complexity status of Independent Set was open until 2004, when Alekseev [2] gave a polynomial time algorithm. Subsequently, Lozin and Milanič [37] gave an algorithm for the *weighted* Indepen-

*Department of Informatics, University of Bergen, Norway, e-mails: `daniello@ii.uib.no`, `yngvev@ii.uib.no`, `vatshelle@ii.uib.no`. Lokshtanov is supported by the Bergen Research Foundation, under the BeHard grant.

DENT SET problem on fork-free graphs. Thus, the only connected five-vertex graph H on which the complexity status of INDEPENDENT SET on H-free graphs remained unknown was the P_5. In this paper we give the first polynomial time algorithm for INDEPENDENT SET on P_5-free graphs, resolving an open problem of [3, 9, 12, 26, 31, 38, 39, 41, 42, 45]. Our algorithm also works for the weighted version of the problem.

Methodology. The starting point of our algorithm is an algorithm of Fomin and Villanger [23] to find large induced subgraphs of constant tree-width. To describe this result we need to introduce some terminology.

A graph H is called *chordal* if it does not contain any cycle on at least four vertices as an induced subgraph. For a graph G a chordal supergraph H of G is called a *triangulation* of G, and a triangulation H is called a *minimal triangulation* if no proper subgraph of H is a triangulation of G. A *clique* in G is a set C of pairwise adjacent vertices in G, and a clique C is called a *maximal clique* if no proper supersets of C are also cliques in G. The tree-width of a graph G is the minimum over all triangulations H of G of the maximum clique size in H, minus 1. It is easy to see that a graph has tree-width 0 if and only if it is an independent set. A set Ω of vertices in G is called a *potential maximal clique* if there exists a minimal triangulation H of G such that Ω is a maximal clique of H.

For every $t \geq 0$, Fomin and Villanger [23] give an algorithm that given as input a graph G together with a list Π of all the potential maximal cliques of G, finds a largest induced subgraph of G of treewidth at most t in time polynomial in $|V(G)|$ and $|\Pi|$. Since independent sets are exactly the graphs of treewidth 0 this algorithm may be used to solve the INDEPENDENT SET problem. The algorithm works in polynomial time on all graph classes where every graph G in the class has at most $|V(G)|^{O(1)}$ potential maximal cliques, and we can list them efficiently. However the graph obtained by taking two cliques of size $n/2$ and joining them by a perfect matching is P_5-free and has $\Omega(2^{n/2})$ potential maximal cliques. Thus, at the first glance, the algorithm above seems useless for our purposes.

A closer inspection of the algorithm of Fomin and Villanger reveals that it produces meaningful output even when given as input a graph G together with a not necessarily exhaustive list Π of potential maximal cliques in G. In fact, the following proposition follows implicitely from the correctness proof of their algorithm.

PROPOSITION 1.1. ([23]) *There is an algorithm that given as input a vertex weighted graph G on n vertices and m edges, together with a list Π of potential maximal cliques, outputs in time $O(|\Pi|n^5m)$ the weight of the maximum weight independent set I such that there exists a minimal triangulation H of G such that every maximal clique C of H is on the list Π and satisfies $|C \cap I| \leq 1$.*

It turns out that for *any* maximal independent set I of a graph G there exists some minimal triangulation H of G such that every maximal clique C of H satisfies $|C \cap I| \leq 1$ (see Lemma 3.1). Thus when Π is an exhaustive list of potential maximal cliques of G, Proposition 1.1 proves the main result of Fomin and Villanger [23] for $t = 0$. When Π is not exhaustive, Proposition 1.1 guarantees that the algorithm will return the weight of some independent set of G.

Furthermore, if we are lucky and G contains a maximum weight independent set I such that there exists a minimal triangulation H of G such that every maximal clique C of H is in Π and satisfies $|C \cap I| \leq 1$, then the algorithm of Proposition 1.1 will output the weight of I. Our main technical contribution is to show that on P_5-free graphs, we can always be lucky. Specifically we will show the following lemma.

LEMMA 1.1. *There is an algorithm that given a P_5-free graph G outputs in time $O(n^9m)$ a family Π of size at most $3n^7$, such that for every maximal independent set I of G with $|I| \geq 2$ there exists an I-good minimal triangulation H of G such that $\zeta(H) \subseteq \Pi$.*

Here $\zeta(H)$ returns the set of maximal cliques of H and an *I-good* minimal triangulation H of G is a minimal triangulation where every vertex $v \in I$ has the same neighbors in G and in H. Any I-good minimal triangulation must satisfy that for every maximal clique C of H, $|C \cap I| \leq 1$. If some clique C contains two vertices of I then they are adjacent in H but not in G, contradicting that H is I-good. Hence feeding the output of Lemma 1.1 directly into the algorithm of Proposition 1.1 yields a $O(n^{12}m)$ time algorithm to compute the weight of the maximum weight independent set, over all independent sets of size at least two. Comparing the output of this algorithm with the weight of the heaviest vertex and selecting the heaviest of the two yields the proof of our main theorem.

THEOREM 1.1. *There is a $O(n^{12}m)$ time algorithm for* WEIGHTED INDEPENDENT SET *on P_5-free graphs.*

Organization of the paper. In Section 2 we give all the necessary definitions and state the known results about minimal triangulations and potential maximal cliques that will be used in the proof. In Section 3 we give a proof of Lemma 1.1. Since Proposition 1.1 is only implicitely proved by Fomin and Villanger [23], we provide for the sake of completeness a proof of a weaker variant of this proposition in Section 4. This variant of Proposition 1.1 is sufficient for an $O(n^{18}m)$ time algorithm for WEIGHTED INDEPENDENT SET on P_5-free graphs. We conclude with some closing remarks in Section 5.

2 Preliminaries

In this paper we deal with graphs that are simple, finite and undirected. For a graph $G = (V, E)$ the integers n and m are used to denote the number of vertices and edges, i.e. $|V| = n$ and $|E| = m$. The *neighborhood* of a vertex v in a graph G is $N_G(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood* of v is $N_G[v] = N(v) \cup \{v\}$. For a vertex set $W \subseteq V(G)$ we similarly define $N_G[W] = \bigcup_{v \in W} N_G[v]$ and $N_G(W) = N_G[W] \setminus W$. To simplify notation we also define for a pair u, v of vertices $N_G(u, v) = N_G(\{u, v\})$ and $N_G[u, v] = N_G[\{u, v\}]$ and $\delta_G(v)$ as $\{u \in N_G(v) : uw \in E$ for some $w \notin N_G[v]\}$. In particular $\delta_G(v)$ are the vertices in $N_G(v)$ with neighbours outside of $N_G[v]$.

For any non-empty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$ and for ease of notation $G \setminus W$ is used for the induced subgraph $G[V \setminus W]$. A *clique* W of a graph G is a subset of V such that all the vertices of W are pairwise adjacent, and an *independent* set is a vertex set W where all vertices are pairwise non-adjacent. For a graph G, let $\zeta(G)$ denote the family of maximal cliques of G.

A path is a sequence of vertices $(w_1, w_2, \ldots, w_\ell)$ such that $w_i w_{i+1} \in E$ for $i \in [1..\ell - 1]$, and the path is called induced if these are the only edges of $G[\{w_1, w_2, \ldots, w_\ell\}]$. If $w_1 w_\ell \in E$ for a path $(w_1, w_2, \ldots, w_\ell)$ then this path is called a cycle and the cycle is called induced if the path becomes induced when removing the edge $w_1 w_\ell$. Induced paths and cycles are also known as chordless paths and cycles.

A vertex set $S \subset V$ of a graph $G = (V, E)$ is called a *separator* if there exist vertices u, v in different connected components of $G \setminus S$ that belong to the same connected component of G. The separator S is called *minimal* for a pair of vertices u, v if no proper subset of S separates u and v. A vertex set S of G is called a minimal separator of G if it is minimal for some pair of vertices in G.

A connected component C of $G \setminus S$ is a *full component* associated with S if $N(C) = S$. The pair (C, S) where C is a full component associated with the minimal separator S is called a *block*. The following proposition is an exercise in [27].

PROPOSITION 2.1. (FOLKLORE) *A set S of vertices of G is a minimal a, b-separator if and only if a and b are in different full components associated with S. In particular, S is a minimal separator if and only if there are at least two distinct full components associated with S.*

For a vertex set $\Omega \subseteq V(G)$ let $\mathcal{C}(\Omega) = C_1, \ldots C_t$ be the connected components of $G \setminus \Omega$. We define the set $\Delta(\Omega)$ to be the family $\{N(C_i) : i \leq t\}$

2.1 Minimal triangulations and chordal graphs

A graph $H = (V, E \cup F)$ is called a *triangulation* of graph $G = (V, E)$ if every cycle of length at least four in H has an *chord*, i.e. an edge between two non consecutive vertices of the cycle. The edges of F, i.e the edges in $E(H) \setminus E(G)$ are called *fill edges*. Triangulated graphs are also called *chordal* graphs. A triangulation $H = (V, E \cup F)$ of a graph G is called minimal if $(V, E \cup F')$ is not chordal for every set $F' \subset F$. Minimal triangulations can be computed in $O(nm)$ time by a range of algorithms, see [29] for a survey.

It is well known that chordal graphs have at most n maximal cliques and at most $n - 1$ minimal separators, that all are cliques [20]. Both the minimal separators and the maximal cliques of a chordal graph can be enumerated in linear time [8].

The *potential maximal cliques* of a graph G are defined to be the set of maximal cliques over all minimal triangulations of the graph G. We denote by $\Omega(G)$ the set of all potential maximal cliques in G. If Ω is a potential maximal clique of G then $\Delta(\Omega)$ will return all minimal separators of G that are subsets of Ω (see [10]).

The following result about the structure of potential maximal cliques is due to Bouchitté and Todinca.

PROPOSITION 2.2. ([10]) *Let $\Omega \subseteq V$ be a set of vertices of the graph G. Then Ω is a potential maximal clique of G if and only if:*

1. *$G \setminus \Omega$ has no full component associated to Ω, i.e. for every $S \in \Delta(\Omega)$ we have $S \subset \Omega$, and*
2. *the graph on the vertex set Ω obtained from $G[\Omega]$ by completing each $S \in \Delta(\Omega)$ into a clique, is a complete graph. In other words, every pair of non-adjacent vertices of Ω is in some $S \in \Delta(\Omega)$.*

Moreover, if Ω is a potential maximal clique, then $\Delta(\Omega)$ is the set of minimal separators of G contained in Ω.

A direct consequence is the following.

PROPOSITION 2.3. ([10]) *There is an algorithm that, given a graph $G = (V, E)$ and a set of vertices $\Omega \subseteq V$, verifies if Ω is a potential maximal clique of G in time $\mathcal{O}(nm)$.*

The following proposition provides a useful tool for manipulating minimal triangulations.

PROPOSITION 2.4. ([10, 34]) *Let X be either a potential maximal clique or a minimal separator of G, and let G_X be the graph obtained from G by completing X into a clique. Then a graph H obtained from G_X by adding a set of fill edges F is a minimal triangulation of G if and only if $F = \bigcup_{C \in \mathcal{C}(X)} F_C$, where F_C is the set of fill edges in a minimal triangulation of $G_X[N[C]]$.*

An immediate consequence of Proposition 2.4 is that a minimal separator or potential maximal clique that already is a clique G remains a minimal separator or potential maximal clique in every minimal triangulation of the graph. Furthermore any minimal triangulation of the graph G can be obtained by combining minimal triangulations of $G[N[C]]$ for each $C \in \mathcal{C}(X)$. Both the minimal separators and the full components associated with them can be mapped back to minimal separators and full components of G.

PROPOSITION 2.5. ([34, 43]) *Let H be a minimal triangulation of G. Then every minimal separator in H is a minimal separator in G.*

PROPOSITION 2.6. ([34, 43]) *Let H be a minimal triangulation of G. Then every full component C associated with a minimal separator S in H is also a full component associated with the minimal separator S in G.*

A *tree decomposition* of a graph G is a pair (T, χ) consisting of a tree T and a function $\chi : V(T) \to 2^{V(G)}$ satisfying the following properties. For every $uv \in E(G)$, $\{u, v\} \subseteq \chi(v)$ for some $v \in V(T)$; and for every vertex $v \in V(G)$ the set $\{u \in V(T) \ : \ v \in \chi(u)\}$ is non-empty and induces a connected subtree of T. The elements of the range $\{\chi(v) \ : \ v \in V(T)\}$ of χ are called *bags* of T. In a *rooted* tree-decomposition (T, χ) the tree T is rooted, and the root vertex of T is denoted by $r(T)$. Tree decompositions are strongly related to chordal graphs due to the following proposition.

PROPOSITION 2.7. ([15, 25, 49]) *A graph G is chordal if and only if there exists a tree decomposition (T, χ) of G such that every bag is a maximal clique in G.*

Such a tree decomposition is referred to as a *clique tree* of the chordal graph G. It is well known that a clique tree of a chordal graph on n vertices and m edges can be constructed in $O(n + m)$ time [8]. We also need the following result relating edges of a clique tree of a chordal graph and its minimal separators.

PROPOSITION 2.8. ([15, 30]) *Let (T, χ) be a clique tree of a chordal graph G. Then S is a minimal separator of G if and only if $S = \chi(u) \cap \chi(v)$ for some edge $uv \in E(T)$.*

3 Enumerating potential maximal cliques

For an independent set I we will say that a triangulation H of G is *I-good* if every vertex v in I satisfies $N_G[v] = N_H[v]$. In other words a triangulation is I-good if no vertex in I is incident to a fill edge.

LEMMA 3.1. *For every graph G and independent set I there exists an I-good minimal triangulation H.*

Proof. Consider the graph $\hat{H}$ obtained from G by turning $V(G) \setminus I$ into a clique. The graph $\hat{H}$ is a split graph, since its vertex set may be partitioned into an independent set (I) and a clique $(V(G) \setminus I)$. Every split graph is chordal, and so $\hat{H}$ is an I-good triangulation of G. There exists a minimal triangulation H such that $G \subseteq H \subseteq \hat{H}$, since $\hat{H}$ is I-good, so is H.

In the remainder of the section, unless explicitely stated otherwise, let G be a P_5 free graph, I be an independent set in G. The aim of this section is to design a polynomial time algorithm that given G outputs a family Π of vertex sets of G such that $|\Pi|$ is polynomial in n and there exists some I-good minimal triangulation H of G such that $\zeta(H) \subseteq \Pi$. We build Π in two steps, specifically $\Pi = \Pi_1 \cup \Pi_2$ where Π_1 and Π_2 are designed to handle different kinds of maximal cliques $\Omega \in \zeta(H)$. We start by defining Π_1.

DEFINITION 1. *For every pair $u, v \in V(G)$ such that $uv \notin E(G)$ define the graph G_{uv} as the graph obtained from G by making $\delta_G(u)$ and $\delta_G(v)$ into cliques, and let H_{uv} be a $\{u,v\}$-good minimal triangulation of G_{uv}. Define the family Π_1 as follows.*

$$\Pi_1 = \bigcup_{\substack{u,v \in V(G), \\ uv \notin E(G)}} \zeta\left(H_{uv}\left[N_G[u, v]\right]\right)$$

OBSERVATION 1. *$|\Pi_1| \leq n^3$ and Π_1 can be computed from G in time $O(n^6)$.*

Proof. The size bound follows from the fact that there are less than n^2 choices for u and v, and for each choice H_{uv} has at most n maximal cliques [20]. For each choice of u and v we can construct G_{uv} in time $O(n^2)$ and a $\{u, v\}$-good triangulation $H'_u v$ obtained from G_{uv} by making all vertices of $V(G_{uv}) \setminus \{u, v\}$ into a clique. Now a *minimal* triangulation H_{uv} of G_{uv} such that $H_{uv} \subseteq H'_{uv}$ can be computed in time $O(n^4)$ by an algorithm of Blair et al. [7]. Since H'_{uv} is $\{u, v\}$-good, so is H_{uv}.

LEMMA 3.2. *For every I-good triangulation H of G and $v \in I$, $H[\delta_G(v)]$ is a clique.*

Proof. Suppose for contradiction that $H[\delta_G(v)]$ is not a clique and assume that two vertices $u, w \in \delta_G(v)$ are non-adjacent in H. Since $u, w \in \delta_G(v)$, u has a neighbor u' (in G) outside $N_G[u]$. Similarly w has a neighbor w' (in G) outside $N_G[u]$. Let C_u be the connected component of $H \setminus \delta_G(v)$ that contains u' and C_w be the connected component of $H \setminus \delta_G(v)$ that contains w'. Note that $u \in N_H(C_u)$ and $w \in N_H(C_w)$. If $w \in N_H(C_u)$ then H is not chordal, since an induced cycle of length at least 4 can be obtained by going from u to v to w and then back to u through C_u. Thus we have $w \notin N_H(C_u)$. An identical argment yields $u \notin N_H(C_w)$. But then C_u and C_w are distinct connected components of $H \setminus \delta_G(v)$ and hence u', u, v, w, w' induces a P_5 in G, yielding the desired contradiction.

LEMMA 3.3. *Let H be an I-good triangulation of G, $u, v \in I$ and C be a connected component of $G \setminus N_G[u, v]$ Then $H[N_G(C)]$ is a clique.*

Proof. Suppose for contradiction that $H[N_G(C)]$ is not a clique and let $u' \in N_G(C)$, $v' \in N_G(C)$ such that $u'v' \notin E(H)$. Since H is I-good and every vertex in $N_G(C)$ has a neighbor in C we have that $N_G(C) \subseteq \delta_G(u) \cup \delta_G(v)$. Lemma 3.2 yields that $H[\delta_G(u)]$ and $H[\delta_G(v)]$ are both cliques. Hence one of u' and v' must be in $\delta_G(u) \setminus \delta_G(v)$ and the other in $\delta_G(v) \setminus \delta_G(u)$. Without loss of generality $u' \in \delta_G(u) \setminus \delta_G(v)$ and $v' \in \delta_G(v) \setminus \delta_G(u)$. Let P be a shortest path in G starting in u', ending in v' and having all internal vertices in C. Such a path exists and has at least one internal vertex, since $u'v' \notin E(G)$. But then u, u', P, v', v is an induced path on at least 5 vertices in G.

Lemma 3.3 combined with well-known facts about clique separators and minimal triangulations yields the following corollaries.

COROLLARY 3.1. *Let H be an I-good minimal triangulation of G, $u, v \in I$ and C be a connected component of $G \setminus N_G[u, v]$. Then $N_H(C) = N_G(C)$ and $H[N_G[C]]$ is a minimal triangulation of $G_{uv}[N_G[C]]$.*

Proof. By Lemma 3.3 $N_G(C)$ is a clique in any I-good triangulation H of G, and hence H is an I-good minimal triangulation of G_{uv}. The vertex set $N_G(C)$ separates C from $V \setminus N[C]$ in the graph G_{uv} and by Proposition 2.4 no fill edges of H can go between different components of $G_{uv} \setminus N_G(C)$. Furthermore, by Proposition 2.4 any minimal triangulation of G_{uv} can be obtained by taking the union of minimal triangulations of $G_{uv}[N[C']]$ for each connected component C' of $G \setminus N_G(C)$.

COROLLARY 3.2. *Let H be an I-good minimal triangulation of G, $u, v \in I$, $C_1, \dots C_t$ be the connected components of $G \setminus N[u, v]$ and Ω be a maximal clique of H. Then Ω is a maximal clique of $H[N_G[u, v]]$ or a maximal clique of $H[N_G[C_i]]$ for some $i \leq t$.*

Proof. By Corollary 3.1 there are no edges in H between vertices of C and $V \setminus N_G[C]$. As there are no edges between these two vertex sets there is also no maximal clique containing vertices of both C and $V \setminus N_G[C]$.

COROLLARY 3.3. *$H_{uv}[N_G[u, v]]$ is a minimal triangulation of $G_{uv}[N_G[u, v]]$.*

Proof. By Corollary 3.1 we have that for every connected component C of $G \setminus N_G[u, v]$, a minimal triangulation H_{uv} of G_{uv} is composed of a minimal triangulation of $G_{uv}[N_G[C]]$ and a minimal triangulation of $G_{uv}[N_G[V \setminus C]]$. Repeating this argument for each connected component C of $V(G) \setminus N_G[u, v]$ until $G_{uv}[N_G[u, v]]$ remains proves the claim.

LEMMA 3.4. *Let Π be a family of subsets of $V(G)$ such that $\Pi_1 \subseteq \Pi$ and let H be an I-good minimal triangulation of G that minimizes $|\zeta(H) \setminus \Pi|$. Let $u, v \in I$ and $\Omega \in \zeta(H)$ such that $\Omega \subseteq N_G[u, v]$. Then $\Omega \in \Pi$.*

Proof. Suppose for contradiction that $\Omega \notin \Pi$. We make a new graph H' as follows: Start with G and for every fill edge xy of H such that $x \notin N_G[u, v]$, add xy to $E(H')$. For every fill edge xy of H_{uv} such that $\{x, y\} \subseteq N_G[u, v]$ add xy to H'. In other words,

$$H' = \big(V(G), E(H_{uv}[N_G[u, v]]) \cup \{xy \in E(H) \ : \ x \notin N_G[u, v]\}\big).$$

We show that H' is an I-good minimal triangulation of G. Let $C_1, \dots C_t$ be the connected components of $G \setminus N_G[u, v]$. We first argue that H' is chordal.

Suppose for contradiction that H' has a chordless cycle Q of length at least 4. If $Q \cap C_i \neq \emptyset$ for some $i \leq t$ there are two cases; either $Q \subseteq N_G[C_i]$ or not. However $Q \subseteq N_G[C_i]$ gives a contradiction, since Lemma 3.3 applied to H_{uv} implies that $H'[N_G(C_i)]$ is a clique, but then $H'[N_G[C_i]] = H[N_G[C_i]]$ and H is chordal. Thus $Q \setminus N_G[C_i] \neq \emptyset$. By Corollary 3.1 we have that $N_{H'}(C_i) = N_H(C_i) = N_G(C_i)$ and hence $N_G(C_i)$ separates C_i from $V(H') \setminus (C_i \cup N_G(C_i))$ in H'. Since Q is a cycle with non-empty intersection both with C_i and with $V(H') \setminus (C_i \cup N_G(C_i))$ it follows that $Q \cap N_G(C_i)$ contains two vertices x and y that are not consecutive on the cycle. But $H'[N_G(C_i)] = H[N_G(C_i)]$ is a clique by Lemma 3.3 and so x and y are adjacent, contradicting that Q is a chordless cycle in H'. We conclude that $Q \cap C_i = \emptyset$ for every i. But then $Q \subseteq N_G[u, v]$ while $H'[N_G[u, v]] = H_{uv}[N_G[u, v]]$, contradicting that H_{uv} is chordal.

Next we argue that H' is I-good. Every fill edge e of H' is either a fill edge of H or a fill edge if $H_{uv}[N_G[u,v]]$. No fill edges of H are incident to I. If e is a fill edge of $H_{uv}[N_G[u,v]]$ then e has both endpoints in $N_G[u,v]]$, and $I \cap N_G[u,v] = \{u,v\}$. Since H_{uv} is $\{u,v\}$-good, e is neither incident to u nor to v, hence H' is I-good.

Finally we argue that H' is a minimal triangulation of G. To that end we use the result of Rose et al. [46] that states that a triangulation $\hat{H}$ of G is a minimal triangulation if and only if $\hat{H} \setminus e$ is not chordal for every $e \in (E(\hat{H}) \setminus E(G))$. Suppose for contradiction that $H' \setminus xy$ is chordal for some edge $xy \in E(H') \setminus E(G)$. If $H' \setminus xy$ is not a supergraph of G_{uv} this contradicts the statement of Lemma 3.2 that every I-good triangulation of G makes $\delta_G(u)$ and $\delta_G(v)$ into cliques. If there exists an $i \leq t$ such that $\{x,y\} \subseteq N_G[C_i]$, then observe that $H'[N_G[C_i]] \setminus xy = H[N_G[C_i]] \setminus xy$. Since $H'[N_G[C_i]] \setminus xy$ is a supergraph of $G_{uv}[N_G[C_i]]$ this implies that $H[N_G[C_i]] \setminus xy$ is a chordal supergraph of $G_{uv}[N_G[C_i]]$, contradicting the conclusion of Corollary 3.1 that $H[N_G[C_i]]$ is a *minimal* triangulation of $G_{uv}[N_G[C_i]]$. Thus $\{x,y\} \subseteq N_G[u,v]$. But then $H'[N_G[u,v]] \setminus xy = H_{uv}[N_G[u,v]] \setminus xy$ is a chordal supergraph of $G_{uv}[N_G[x,y]]$ contradicting the conclusion of Corollary 3.3 that $H_{uv}[N_G[u,v]]$ is a minimal triangulation of $G_{uv}[N_G[x,y]]$. Hence H' is an I-good minimal triangulation of G.

By Corollary 3.2 every maximal clique Ω' of H' is a maximal clique of $H'[N_G[u,v]]$ or a maximal clique of $H'[N_G[C_i]]$. In the case that Ω' is a maximal clique of $H'[N_G[u,v]] = H_{uv}[N_G[u,v]]$, we have $\Omega' \in \Pi_1$. Consider now the case that Ω' is a maximal clique of $H'[N_G[C_i]]$ for some i but not a maximal clique of $H'[N_G[u,v]]$. In this case Ω' contains at least one vertex of C_i. Further, by Lemma 3.3 $H'[N_G[C_i]] = H[N_G[C_i]]$ and hence Ω' is a maximal clique of H. This implies that every maximal clique Ω' of H' that is not a maximal clique of H is in Π. Further, Ω is a maximal clique of H, $\Omega \notin \Pi$. Since $\Omega \subseteq N_G[u,v]$ and every maximal clique of H' which is a subset of $N_G[u,v]$ is in Π, Ω is *not* a maximal clique of H'. Hence $(\zeta(H') \setminus \Pi) \setminus \zeta(H) = \emptyset$, while $(\zeta(H) \setminus \Pi) \setminus \zeta(H') \neq \emptyset$, implying $|\zeta(H) \setminus \Pi| > |\zeta(H') \setminus \Pi|$ and contradicting the choice of H.

Armed with Lemma 3.4, the next natural goal is to find a polynomial size family Π_2 that will contain every maximal clique Ω of an I-good minimal triangulation H such that Ω is not a subset of $N_G[u,v]$ for any choice of $u,v \in I$. We will compute such a set Π_2 in an indirect way. We will first define a polynomial size family Δ_2 of vertex sets of G such that any Ω that is not a subset of $N_G[u,v]$ for any choice of $u,v \in I$ satisfies $\Delta(\Omega) \subseteq \Delta_2$. We then show the following result.

Lemma 3.5. *There is an algorithm that given as input a P_5-free graph G and family Δ of vertex sets in G, outputs in time $O(|\Delta|n^6m)$ the family*

$$\{\Omega \in \Omega(G) \ : \ \Delta(\Omega) \subseteq \Delta\}.$$

Further, the size of the family output by the algorithm is at most $O(2|\Delta|n^4)$.

The proof of Lemma 3.5 is postponed to Section 3.1. We will then define $\Pi_2 = \{\Omega \in \Omega(G) \ : \ \Delta(\Omega) \subseteq \Delta_2\}$. Thus we may compute Π_2 from Δ_2 in polynomial time using Lemma 3.5. We start by defining Δ_2.

Definition 2. *Let Δ_2 be a set of vertex sets such that $N_G(\hat{C}_u) \in \Delta_2$ for each ordered triple (u,v,w) of vertices where:*

- *$\{u,v,w\}$ is an independent set in G,*
- *C_w is the connected component of $G \setminus N_G[u,v]$ containing w, and*
- *$\hat{C}_u$ is the connected component of $G \setminus N_G[C_w]$ containing u.*

Observation 2. *There is an algorithm that given G outputs Δ_2 in time $O(n^4m)$. Furthermore, $|\Delta_2| \leq n^3$.*

Proof. $|\Delta_2| \leq n^3$ follows from the fact that each set in Δ_2 is uniquely defined by the three vertices u, v and w. The algorithm to compute Δ_2 goes over all possible choices for u, v and w, computes C_w using a BFS from w in $O(n+m)$ time, then computes $\hat{C}_u$ by a BFS from u in $O(n+m)$ time.

We remark that the running time of the algorithm computing Δ_2 can be improved, however this does not affect the running time of our final algorithm for Independent Set on P_5-free graphs.

We aim to zero in on the maximal cliques of I-good minimal triangulations that cannot be covered by the neighborhood of two I-vertices. We prove a few simple properties of such maximal cliques.

Lemma 3.6. *Let I be a maximal independent set of G with $|I| \geq 2$, H be an I-good minimal triangulation of G and $\Omega \in \zeta(H)$ be such that there is no $u,v \in I$ with $\Omega \subseteq N_G[u,v]$ then*

1. *$\Omega \cap I = \emptyset$,*
2. *Every component C of $G \setminus \Omega$ contains a vertex of I,*
3. *No set $S \in \Delta(\Omega)$ contains all other sets in $\Delta(\Omega)$.*

Proof. To see that $\Omega \cap I = \emptyset$, observe that Ω is a clique in H and hence if $u \in I \cap \Omega$ it follows that $\Omega \subseteq N_H[u]$. But H is I-good and hence $N_H[u] = N_G[u]$. Let v be an arbitrary vertex of $I \setminus \{u\}$, then $\Omega \subseteq N_G[u, v]$, a contradiction.

If some component C of $G \setminus \Omega$ is disjoint from I, let u be an arbitrary vertex of C. Since $\Omega \cap I = \emptyset$, the set $I \cup \{u\}$ is independent in G, contradicting that I is a maximal independent set of G.

Suppose for contradiction that there exists an $S \in \Delta(\Omega)$ such that for every $S' \in \Delta(\Omega)$, $S' \subseteq S$. By Proposition 2.2 $\Omega \setminus S \neq \emptyset$. Let $u \in \Omega \setminus S$, since $\Omega \cap I = \emptyset$ the set $I \cup \{u\}$ is independent in G, contradicting that I is a maximal independent set of G.

Lemma 3.7. *A graph G is P_5-free if and only for every pair u, v of non-adjacent vertices and every minimal u, v-separator S, we have that $S \subseteq N_G(u, v)$.*

Proof. For the forward direction, assume that G is P_5-free, let u, v be non-adjacent vertices in G and S be a minimal u, v-separator. Assume for a contradiction that S is not a subset of $N_G(u, v)$, hence there is a node $w \in S$ which is neither adjacent to u nor to v. Let C_u and C_v be the connected components of $G \setminus S$ containing u and v respectively. Since S is a minimal separator w has neighbors in both C_u and C_v, and hence both $G[C_u \cup \{w\}]$ and $G[C_v \cup \{w\}]$ are connected. Let P_u be a shortest path from u to w in $G[C_u \cup \{w\}]$ and P_v be a shortest path from v to w in $G[C_v \cup \{w\}]$. Both P_u and P_v have at least 3 vertices (including w) since neiter u nor v are adjacent to w. Thus $G[V(P_u) \cup V(P_v)]$ contains an induced P_5, contradicting that G is P_5-free. We conclude that $S \subseteq N(u, v)$.

Now assume that G is not P_5-free, and let v_1, v_2, v_3, v_4, v_5 be an induced P_5 in G. Set $S^* = V(G) \setminus \{v1, v2, v4, v5\}$. We have that S^* is a v_1, v_5-separator. Let $S \subseteq S^*$ be a minimal v_1, v_5 separator. Now $v_3 \in S$ since otherwise v_1, v_2, v_3, v_4, v_5 is a path in $G \setminus S$, contradicting that S is a v_1, v_5 separator. Further v_3 is neither adjacent to v_1 nor v_5 and so S is a minimal v_1, v_5 separator which is not a subset of $N(v_1, v_5)$.

We will also need the following simple lemma about potential maximal cliques in any graph G.

Lemma 3.8. *Let G be a graph, Ω be a potential maximal clique of G, C_u and C_v be components of $G \setminus \Omega$ such that $N_G(C_v) \setminus N_G(C_u) \neq \emptyset$. Let $u \in C_u$, $v \in C_v$. Then $N_G(C_u)$ is a minimal u-v-separator.*

Proof. Clearly $N(C_u)$ separates u from v. We now show that $N(C_u)$ is a *minimal* u-v-separator, specifically that for any $x \in N_G(C_u)$ there is a path from u to v in $G \setminus (N_G(C_u) \setminus \{x\})$. Clearly there is a path from u to x. Let $y \in N_G(C_v) \setminus N_G(C_u)$. If $xy \in e$ then there is a path from x to y via the edge xy and then to v through C_v. If $xy \notin E(G)$, Proposition 2.2 implies that there is a component C of $G \setminus \Omega$ such that $\{x, y\} \subseteq N_G(C)$. In this case there is a path from u to x (through C_u), from x to y (through C) and finally from y to v (through C_v).

Lemma 3.9. *Let I be a maximal independent set of G with $|I| \geq 2$, H be an I-good minimal triangulation of G and $\Omega \in \zeta(H)$ be such that there is no $u, v \in I$ with $\Omega \subseteq N_G[u, v]$ then $\Delta(\Omega) \subseteq \Delta_2$*

Proof. Let $S_u \in \Delta(\Omega)$, we argue that $S_u \in \Delta_2$. Let $\hat{C}_u$ be a connected component of $G \setminus \Omega$ such that $N_G(\hat{C}_u) = S_u$. By Lemma 3.6 $\hat{C}_u \cap I \neq \emptyset$, so let u be a vertex in $\hat{C}_u \cap I$. By Lemma 3.6 $G \setminus \Omega$ has a connected component $\hat{C}_v$ such that $N_G(\hat{C}_v) \setminus N_G(\hat{C}_u) \neq \emptyset$. If $N_G(\hat{C}_u) \subseteq N_G(\hat{C}_v)$ then, by Proposition 2.2 there is a vertex $w \in \Omega \setminus (N_G(C_u) \cup N_G(C_v)) = \Omega \setminus N_G(C_v)$. Such a w satisfies $w \notin N_G[u, v]$. If on the other hand $N_G(\hat{C}_u) \setminus N_G(\hat{C}_v) \neq \emptyset$ then Lemma 3.8 implies that both $N_G(\hat{C}_u)$ and $N_G(\hat{C}_v)$ are minimal u-v-separators. Then, by Lemma 3.7 $N_G(\hat{C}_u) \cup N_G(\hat{C}_v) \subseteq N_G(u, v)$. By choice of Ω there must be a vertex $w \in \Omega \setminus N_G[u, v]$. In either case we selected a vertex $w \in \Omega \setminus N_G[u, v]$ such that $w \notin N_G(\hat{C}_u) \cup N_G(\hat{C}_v)$. Let C_w be the component of $G \setminus N_G[u, v]$ that contains w. We show that u, v and w witness that $S_u = N(\hat{C}_u) \in \Delta_2$. Specifically we argue that $\hat{C}_u$ is a connected component of $G \setminus N_G[C_w]$. To that end we need two argue that (a) $N_G[C_w] \cap \hat{C}_u = \emptyset$ and (b) that $N_G(\hat{C}_u) \subseteq N_G[C_w]$.

We first show that $N_G[C_w] \cap \hat{C}_u = \emptyset$. Suppose not, then $C_w \cap N_G[\hat{C}_u] \neq \emptyset$. Furthermore, $w \notin N_G[\hat{C}_u]$ and hence $C_w \setminus N_G[\hat{C}_u] \neq \emptyset$. Since $G[C_w]$ is connected this implies that $C_w \cap N_G(\hat{C}_u) \neq \emptyset$. However $N_G(\hat{C}_v) \setminus N_G(\hat{C}_u) \neq \emptyset$ and therefore Lemma 3.8 yields that $N_G(\hat{C}_u)$ is a minimal u-v-separator. Thus, by Lemma 3.7, $N_G(\hat{C}_u) \subseteq N_G(u, v)$. But $N_G(u, v) \cap C_w = \emptyset$, a contradiction to $C_w \cap N_G(\hat{C}_u) \neq \emptyset$.

We now show that $N_G(\hat{C}_u) \subseteq N_G[C_w]$. For each $x \in N_G(\hat{C}_u)$ we have $x \in \Omega$. Hence, if $wx \notin E(G)$ Proposition 2.2 implies that there is a component C of $G \setminus \Omega$ such that $\{w, x\} \subseteq N_G(C)$. We argue that $C \subseteq C_w$, this implies that $x \in N_G[C_w]$. Suppose that $C \setminus C_w \neq \emptyset$, then, since $G[C]$ is connected $C \cap N_G(C_w) \neq \emptyset$. But $N_G(C_w) \subseteq N_G(u, v)$ and therefore C must either contain a neighbor of u or a neighbor of v. But then $C = \hat{C}_u$ or $C = \hat{C}_v$ which contradicts that $w \notin N_G(C_u) \cup N_G(C_v)$ while $w \in N(C)$.

We are now in position to prove Lemma 1.1.

Lemma 1.1 (restated). *There is an algorithm that given a graph P_5-free graph G outputs in time $O(n^9 m)$ a family Π of size at most $3n^7$, such that for every maximal independent set I of G with $|I| \geq 2$ there exists an I-good minimal triangulation H of G such that $\zeta(H) \subseteq \Pi$.*

Proof. The algorithm computes the family Π_1 of size n^3 in time $O(n^3 m)$ using Observation 1. Then it computes the set Δ_2 of size at most n^3 in time $O(n^4 m)$ using Observation 2. Finally it computes the set $\Pi_2 = \{\Omega \in \Omega(G) \ : \ \Delta(\Omega) \subseteq \Delta_2\}$ from Δ_2 in time $O(n^9 m)$ using Lemma 3.5. By Lemma 3.5, $|\Pi_2| \leq 2n^7$. The algorithm outputs the family $\Pi = \Pi_1 \cup \Pi_2$. The size of Π_2 is upper bounded by $3n^7$.

We need to argue that there exists an I-good minimal triangulation H of G such that $\zeta(H) \subseteq \Pi$. Let H be an I-good minimal triangulation of G that minimizes $\zeta(H) \setminus \Pi$. By Lemma 3.4, for every $\Omega \in \zeta(H)$ such that there exists a pair $u, v \in I$ such that $\Omega \subseteq N_G[u, v]$ we have $\Omega \in \Pi$. On the other hand, for every $\Omega \in \zeta(H)$ for which no such pair exists Lemma 3.9 yields that $\Delta(\Omega) \subseteq \Delta_2$. Further, Ω is a potential maximal clique of G and hence $\Omega \in \Omega(G)$. But then $\Omega \in \Pi_2$, concluding the proof.

3.1 From Minimal Separators to Potential Maximal Ciques

The goal of this section is to prove the following result.

Lemma 3.5 (restated). *There is an algorithm that given as input a P_5-free graph G and family Δ of vertex sets in G, outputs in time $O(|\Delta| n^6 m)$ the family*

$$\{\Omega \in \Omega(G) \ : \ \Delta(\Omega) \subseteq \Delta\}.$$

Further, the size of the family output by the algorithm is at most $2|\Delta| n^4$.

In order to prove Lemma 3.5 we will employ a few results by Bouchitté and Todinca [11] on the enumeration of potential maximal cliques. Let Ω be a potential maximal clique of a graph G and let $S \in \Delta(\Omega)$. We say that S is *active for* Ω if Ω is not a clique in the graph $G_{\Omega,S}$ obtained from G by completing all sets $S' \in (\Delta(\Omega) \setminus \{S\})$ to cliques. If S is active, a pair of vertices $x, y \in S$ non adjacent in $G_{\Omega,S}$ is called an *active pair*. We say that a potential maximal clique Ω is *nice* if at least one set $S \in \Delta(\Omega)$ is active for Ω. Let $v_1, v_2, \ldots, v_n$ be an arbitrarily chosen vertex ordering of V. Define $V_i = \{v_1, v_2, \ldots, v_i\}$ and $G_i = G[V_i]$. We will say that a potential maximal clique Ω_i is a *seed* if there is a set $S_i \in \Delta_{G_i}(\Omega_i)$ and vertex $v \in V_i$ such that $\Omega_i = S_i \cup \{v\}$, or if Ω_i is a nice potential maximal clique of G_i.

LEMMA 3.10. ([11]) *Let Ω be a potential maximal clique of G. Then there exists an $i \leq n$ such that $\Omega_i = \Omega \cap V_i$ is a seed potential maximal clique of G_i.*

LEMMA 3.11. ([11]) *For every $i \leq n$ and potential maximal clique Ω_i of G_i, there is exactly one potential maximal clique Ω of G such that $\Omega_i = \Omega \cap V_i$. Furthermore, given Ω_i we can compute Ω in $O(n^2 m)$ time.*

Lemma 3.10 and Lemma 3.11 are not stated explicitely in this way by [11]. However, Lemma 3.10 is a direct corollary of Theorem 20 of [11], while Lemma 3.11 follows from Corollary 21 and Corollary 12 of [11]. For every $i \leq n$ we make a set Δ^i from Δ as follows.

DEFINITION 3. *For every $i \leq n$ let Δ^i be a set of vertex sets of G_i such that $N_{G_i}(C_i) \in \Delta^i$ for each pair (S, C_i) where $S \in \Delta$ and C_i is a connected component of $G_i \setminus S$.*

OBSERVATION 3. *For every i $|\Delta_i| \leq |\Delta| n$, and Δ_i can be computed from Δ in time $O(|\Delta| n^2)$.*

Proof. $|\Delta_i| \leq |\Delta| n$ since there are $|\Delta|$ choices for S and at most n components of $G_i \setminus S$. We can compute the connected components of $G \setminus S$ in time $O(n+m)$, and for each component C_i listing $N_{G_i}(C_i)$ takes time at most $O(n)$.

LEMMA 3.12. *Let $\Omega \in \Omega(G)$ be such that $\Delta(\Omega) \subseteq \Delta$. Let $\Omega_i = \Omega \cap V_i$. Then $\Delta_{G_i}(\Omega_i) \subseteq \Delta^i$.*

Proof. Let $S_i \in \Delta_{G_i}(\Omega_i)$ and let C_i be the connected component of $G_i \setminus \Omega_i$ such that $N_{G_i}(C_i) = S_i$. Since $\Omega_i = \Omega \cap V_i$ it follows that $\Omega \cap C_i = \emptyset$ and hence there is a connected component C of $G \setminus \Omega$ such that $C_i \subseteq C$. Since $N_{G_i}(C_i) \subseteq \Omega_i \subseteq \Omega$ it follows that $N_{G_i}(C_i) \subseteq N(C)$. Let $S = N(C)$, since $N_{G_i}(C_i) \subseteq S$ it follows that C_i is a connected component of $G_i \setminus S$. But $S_i = N_{G_i}(C_i) = S_i$ so (S, C_i) witness the fact that $S_i \in \Delta_{G_i}(\Omega_i)$.

DEFINITION 4. *For every $i \leq n$ define*

$$\Pi_A^i = \{(S_i \cup \{v\}) \in \Pi^i \ : \ S_i \in \Delta^i \text{and } v \in V_i\}.$$

DEFINITION 5. *For every $i \leq n$ let Π_B^i be a set of vertex sets of G_i such that $S_i \cup (N_{G_i}(v) \cap C_i) \in \Pi_B^i$ for each triple (S_i, v, C_i) where $S_i \in \Delta^i$, $v \in S_i$ and C_i is a connected component of $G_i \setminus S_i$.*

OBSERVATION 4. *For every $i \leq n$, $|\Pi_A^i| \leq |\Delta| n^2$ and $|\Pi_B^i| \leq |\Delta| n^3$. Furthermore, Π_A^i can be computed in time $O(|\Delta| n^3)$ and Π_B^i in time $O(|\Delta| n^4)$.*

Proof. $|\Pi_A^i| \leq |\Delta| n^2$ and that Π_A^i can be computed in time $O(|\Delta| n^3)$ follows directly from the definition of Π_A^i together with Observation 3. $|\Pi_B^i| \leq |\Delta| n^3$ also follows from Observation 3 since each $\Omega_i \in \Pi_B^i$ is indexed by a set S in Δ^i, a connected component C_i of $G_i \setminus S$ and a vertex in S. For each choice of $S \in \Delta^i$ we can find connected components of $G_i \setminus S$ in time $O(n+m)$. For each C_i and each v we may list out $S_i \cup (N_{G_i}(v) \cap C_i)$ in time $O(n)$, giving the $O(|\Delta| n^4)$ time bound to compute Π_B^i.

LEMMA 3.13. *Let Ω_i be a seed potential maximal clique of G_i such that $\Delta_{G_i}(\Omega_i) \subseteq \Delta^i$. Then $\Omega_i \in (\Pi^i_A \cup \Pi^i_B)$.*

Proof. There are two cases, either Ω_i is nice or not. Suppose Ω_i is not nice. Then, since Ω_i is a seed, there is a set $S_i \in \Delta_{G_i}(\Omega_i)$ and vertex $v \in V_i$ such that $\Omega_i = S_i \cup \{v\}$. But then $S_i \in \Delta^i$ and hence (S_i, v) witnesses that $\Omega_i \in \Pi^i_A$.

Suppose now that Ω_i is nice and let $S_i \in \Delta_{G_i}(\Omega_i)$ be an active set for Ω_i. Let $x, y \in S_i$ be an active pair, that is $xy \notin E(G_i)$ and no $S' \in (\Delta_{G_i}(\Omega_i) \setminus \{S_i\})$ contains both x and y. Let $X_1, \ldots X_t$ be the connected components of $G_i \setminus \Omega_i$ that satisfy $N_{G_i}(X_j) = S_i$, and let $X = \bigcup_{j \le t} X_j$. Let P_{xy} be a shortest path between x and y in $G_i[X_1 \cup \{x, y\}]$, since $xy \notin E(G_i)$, P_{xy} has at least three vertices.

We argue that $N_{G_i}(x) \setminus (X \cup \Omega_i) = \emptyset$ or $N_{G_i}(y) \setminus (X \cup \Omega_i) = \emptyset$. Suppose not, and let $x' \in N_{G_i}(x) \setminus (X \cup \Omega_i)$ and $y' \in N_{G_i}(y) \setminus (X \cup \Omega_i)$. No component of $G_i \setminus \Omega_i$ except for the X_j's is adjacent to both x and y, while the component C_x of $G_i \setminus \Omega_i$ that contains x' is adjacent to x and the component C_y of $G_i \setminus \Omega_i$ that contains y' is adjacent to y. Hence C_x and C_y are distinct components of $G_i \setminus \Omega_i$. But then x', x, P_{xy}, y, y' forms an induced path on at least five vertices in G_i, a contradiction.

Let $v \in \{x, y\}$ be such that $N_{G_i}(v) \setminus (X \cup \Omega_i) = \emptyset$ and let $u \in \Omega_i \setminus S_i$. We argue that $uv \in E(G_i)$. Suppose not, then by Proposition 2.2 there is a connected component C of $G_i \setminus \Omega_i$ such that $\{u, v\} \subseteq N_{G_i}(C)$. But $N(C) \neq S_i$ implies that $C \cap X = \emptyset$ and v has a neighbor $v' \in C$, contradicting that $N_{G_i}(v) \setminus (X \cup \Omega_i) = \emptyset$. Thus v is adjacent to every u in $\Omega_i \setminus S_i$, or in other words $\Omega_i \setminus S_i \subseteq N_{G_i}(v)$.

Since Ω_i is a potential maximal clique of G_i, Proposition 2.2 implies that there is a connected component C_i of $G \setminus S_i$ such that $(\Omega_i \setminus S_i) \subseteq C_i$. We show that $\Omega_i = S_i \cup (N_{G_i}(v) \cap C_i)$.

We have that $\Omega_i \subseteq C_i \cup S_i$ and that $\Omega_i \setminus S_i \subseteq N_{G_i}(v)$, therefore

$$\begin{aligned} S_i \cup (N_{G_i}(v) \cap C_i) &\supseteq S_i \cup (N_{G_i}(v) \cap (\Omega_i \setminus S_i)) \\ &\supseteq S_i \cup (\Omega_i \setminus S_i) \\ &= \Omega_i. \end{aligned}$$

Furthermore the components $X_1, \ldots X_t$ are also connected components of $G_i \setminus S_i$, but each X_j is disjoint from Ω_i and hence $C_i \cap X = \emptyset$. Since $N_{G_i}(v) \setminus (X \cup \Omega_i) = \emptyset$ it follows that $N_{G_i}(v) \cap C_i \subseteq \Omega_i$. This yields $S_i \cup (N_{G_i}(v) \cap C_i) \subseteq \Omega_i$. We conclude that $\Omega_i = S_i \cup (N_{G_i}(v) \cap C_i)$. Thus (S_i, v, C_i) witnesses that $\Omega_i \in \Pi^i_B$, concluding the proof.

We are now ready to prove the main result of this section.

Proof. (*of Lemma 3.5*) Given G and Δ we compute Π^i_A and Π^i_B using Observation 4. For each $\Omega_i \in \Pi^i_A \cup \Pi^i_B$ we check using Proposition 2.3 in time $O(nm)$ whether Ω_i is a potential maximal clique of G_i. If so, we compute in $O(n^2 m)$ time using Lemma 3.11 the unique potential maximal clique Ω such that $\Omega_i = \Omega \cap V_i$. Finally, we check whether $\Delta(\Omega) \subseteq \Delta$, and if so, the algorithm adds Ω to its output family. Checking whether $\Delta(\Omega) \subseteq \Delta$ can be done in time $O(n^2)$; for each of the n sets $S \in \Delta(\Omega)$ we can check whether $S \in \Delta$ if Δ is stored as a prefix-tree. If the input family Δ is given as a list of sets, rather than a prefix tree we may compute a prefix tree for Δ in time $O(|\Delta| n \log n)$ before starting the remaining computation. Thus the total running time of the algorithm is bounded by $O(|\Delta| n^6 m)$ as claimed. The algorithm outputs at most one potential maximal clique Ω for every $i \le n$ and $\Omega_i \in \Pi^i_A \cup \Pi^i_B$. Therefore the family output by the algorithm has size at most $2|\Delta| n^4$.

For every Ω output by the algorithm, Ω is a potential maximal clique of G (by Lemma 3.11) and $\Delta(\Omega) \subseteq \Delta$ since we verify that it is. It remains to show that for every potential maximal clique Ω of G such that $\Delta(\Omega) \subseteq \Delta$, Ω is output by the algorithm. By Lemma 3.10 there is an $i \le n$ such that $\Omega_i = \Omega \cap V_i$ is a seed of G_i. By Lemma 3.13, $\Omega_i \in \Pi^i_A \cup \Pi^i_B$. Since Ω_i is a potential maximal clique of G_i the algorithm will output the unique potential maximal clique of G whose intersection with V_i is Ω_i, namely Ω.

4 Dynamic Programming for Independent set

We prove the following variant of Proposition 1.1.

LEMMA 4.1. *There is an algorithm that given as input a vertex weighted graph G on n vertices and m edges, together with a list Π of potential maximal cliques in G, outputs in time $O(|\Pi|^2 n^4 m)$ the weight of the maximum weight independent set I such that there exists a minimal triangulation H of G such that every maximal clique C of H is on the list Π and satisfies $|C \cap I| \le 1$. If no such independent set exists, the algorithm outputs $-\infty$.*

To prove Lemma 4.1 we prove a slightly more general result, stated in terms of tree-decompositions, rather than triangulations.

DEFINITION 6. (I-SPARSE TREE-DECOMPOSITION) *For an independent set $I \subseteq V(G)$, a tree-decomposition (T, χ) of G is called I-*sparse *if for each bag B we have $|B \cap I| \le 1$.*

DEFINITION 7. (SIMPLE TREE DECOMPOSITION) *Let G be a graph and (T, χ) be a rooted tree decomposition* *of G. We say (T, χ) is simple if (a) no bag B is a subset of any other bag B' and (b) For every $u, v \in V(T)$ where v is a descendant of u in T, there exists a component $C \in \mathcal{C}(\chi(u))$ such that $\chi(v) \subset \chi(u) \cup C$.*

Rooting a clique tree of a minimal triangulation at an arbitrary vertex yields a simple tree decomposition. On the other hand, Proposition 2.4 implies that any simple tree-decomposition of G whose bags are potential maximal cliques of G is a clique-tree of a minimal triangulation H of G. Hence Lemma 4.1 follows directly from the following lemma.

LEMMA 4.2. *There is an algorithm that given as input a vertex weighted graph G on n vertices and m edges, together with a list Π of vertex sets in G, outputs in time $O(|\Pi|^2 n^4 m)$ the weight of the maximum weight independent set I such that there exists an I-sparse simple tree decomposition (T, χ) of G such that $\chi(v) \in \Pi$ for all $v \in V(T)$. If no such independent set exists, the algorithm outputs $-\infty$.*

We now define a function M as follows. M takes as input a vertex set $B \in \Pi$, a vertex set $X \subseteq B$ with $|X| \leq 1$ and a component $C \in \mathcal{C}(B)$. The function returns the weight of the maximum weight independent set $I \subseteq B \cup C$ such that $I \cap B = X$ and there exists an I-sparse simple tree-decomposition (T, χ) of $G[X \cup C]$ such that all bags of (T, χ) are in Π and $\chi(r(T)) = B$ for the root vertex $r(T)$ of T. If no such independent set exists, M returns $-\infty$.

LEMMA 4.3. *For every set $B \in \Pi$, $X \subseteq B$ with $|X| \leq 1$ and component $C \in \mathcal{C}(B)$, M satisfies the following recurrence.*

$$M(B, X, C) = w(X) + \max_{B', X'} \left[w(X' \setminus X) + \sum_{\substack{C' \in \mathcal{C}(B') \\ C' \subseteq C}} (M(B', X', C') - w(X')) \right]$$

Here the maximum is taken over all sets $B' \subseteq \Pi$ and $X' \subseteq B'$ such that (i) $B' \subseteq B \cup C$, (ii) $N(C) \subseteq B'$, (iii) $B' \cap C \neq \emptyset$, (iv) $|X'| \leq 1$, (v) $B \cap B' \cap X = B \cap B' \cap X'$ and (vi) $G[X' \cup X]$ is independent. If no such B' and X' exist the recurrence above is not well defined, and in this case $M(B, X, C) = -\infty$.

Proof. For the $\geq$ direction of the proof, the inequality holds automatically if the right hand side is $-\infty$. Consider therefore a pair B', X' satisfying conditions $(i)-(vi)$. For each $C' \in \mathcal{C}(B')$ let $I_{C'}$ be an independent set of weight $M(B', X', C')$ such that $I_{C'} \cap B' = X'$, and let $(T_{C'}, \chi_{C'})$ be an $I_{C'}$-sparse simple tree-decomposition of $G[B' \cap C']$ such that $\chi(r(T_C)) = B'$. Set

$$I = \bigcup_{\substack{C' \in \mathcal{C}(B'), \\ C' \subseteq C}} I_{C'} \cup X$$

Clearly I is independent. For the weight of I we have that

$$w(I) = w(X) + w(X' \setminus X) + \sum_{\substack{C' \in \mathcal{C}(B') \\ C' \subseteq C}} (M(B', X', C') - w(X')) .$$

We now build a rooted tree-decomposition (T, χ) of $G[B \cup C]$ by identifying all of the tree-decompositions $(T_{C'}, \chi_{C'})$ at their root r, adding a new root vertex r' and making r' the parent of r, and setting $\chi(r) = B$. For all other vertices $v \in V(T)$, v is a vertex of $T_{C'}$ for some $C' \in \mathcal{C}(B')$ with $C' \subseteq C$. We set $\chi(v) = \chi_{C'}(v)$. It is easy to verify that (T, χ) is indeed an I-sparse simple tree-decomposition of $G[B \cup C]$ with all bags from Π, hence $M(B, X, C) \geq w(I)$ proving the $\geq$ direction of the inequality.

For the $\leq$ direction of the equality, the inequality is trivially true if the left hand side is $-\infty$. Thus, let I be an independent set in $G[B \cup C]$ of weight $M(B, X, C)$ such that $I \cap B = X$ and let (T, χ) be an I-good simple tree-decomposition of $G[B \cup C]$ with root bag B. We claim that $r(T)$ only has one child. Suppose not and let v_1 and v_2 be two children of $r(T)$. Since no bag is a subset of another bag and $\chi(r(T)) = B$ it follows that $\chi(v_1) \cap C \neq \emptyset$ and $\chi(v_2) \cap C \neq \emptyset$. But this contradicts that the set $\{v \in V(T) \ : \ \chi(v) \cap C \neq \emptyset\}$ induces a connected subtree of T. Hence $r(T)$ has only one child r'. Let $B' = \chi(r')$ and $X' = I \cap B'$. The sets B' and X' satisfy conditions (i)-(vi). For each $C' \in \mathcal{C}(B')$ such that $C' \subseteq C$ let $I'_C = I \cap (B' \cup C')$. Observe that

$$w(I) = w(X) + w(X' \setminus X) + \sum_{\substack{C' \in \mathcal{C}(B') \\ C' \subseteq C}} (w(I_{C'} - w(X')) .$$

Hence, to complete the proof it is sufficient to prove that for every $C' \in \mathcal{C}(B')$ such that $C' \subseteq C$, $M(B', X', C') \geq w(I_{C'})$. Consider the tree-decomposition (T, χ) and let $Z_{C'}$ be the vertices in T whose bags have non-empty intersection with C', that is $Z_{C'} = \{v \in V(T) \ : \ \chi(v) \cap C' \neq \emptyset\}$. The set $Z_{C'}$ is a connected subtree of T. Furthermore, since (T, χ) is simple, for every vertex $v \in Z_{C'}$, all vertices of $\chi(v)$ are in the same connected component of $G \setminus B'$, hence $\chi(v) \subseteq C'$. Thus $(T[Z_{C'}], \chi)$ (with χ restricted to $Z_{C'}$) is an $(I_{C'} \cap C')$-sparse simple tree decomposition of $G[C']$ with all bags from Π. Let

$r^\star$ be the vertex in $Z_{C'}$ closest to the root of T. We argue that $N(C') \subseteq \chi(r^\star)$. For each $a \in N(C')$ there is some $b \in C'$ such that $ab \in E(G)$. The topmost bag in T that contains b is $\chi(r^\star)$ or $\chi(u)$ for a descendant u of $r^\star$. Furthermore $a \in B'$. Since some bag of T must contain the edge ab and the set of bags containing a is connected in T, it follows that $a \in \chi(r^\star)$. Hence $N(C') \subseteq \chi(r^\star)$. We make a tree-decomposition $(T_{C'}, \chi_{C'})$ from $(T[Z_{C'}], \chi)$ by attaching a new node $\hat{r}$ to $r^\star$, making $\hat{r}$ the parent of $r^\star$ in T, setting $\chi_{C'}(\hat{r}) = B'$ and $\chi_{C'}(v) = \chi(v)$ for all $v \in Z_{C'}$. The tree-decomposition $(T_{C'}, \chi_{C'})$ is an $I_{C'}$-sparse simple tree-decomposition of $G[B' \cup C']$ using only bags from Π. Thus $M(B', X', C') \geq w(I_{C'})$, completing the proof.

We conclude the section with a proof of Lemma 4.2.

Proof. (*of Lemma 4.2*) The weight of the maximum weight independent set I such that there exists an I-sparse simple tree decomposition (T, χ) of G is exacly

$$\max_{B,X} \left[w(X) + \sum_{C \in \mathcal{C}(C)} (M(B, X, C) - w(X)) \right] \tag{4.1}$$

where the maximum is taken over all $B \in \Pi$ and $X \subseteq B$ with $|X| \leq 1$.

The algorithm computes $M(B, X, C)$ for every choice of B, X and C using the recurrence of Lemma 4.3. In order to compute the value of $M(B, X, C)$ using this recurrence we only need to look up the value for $M(B', X', C')$ for choices of (B', X', C') with $|C'| < |C|$. Thus we process the triples (B, X, C) sorted by $|C|$. Once the algorithm has pre-computed the value of $M(B, X, C)$ for every possible choice of B, X and C, it computes the weight of the maximum weight independent set I such that there exists an I-sparse simple tree decomposition (T, χ) of G using Equation 4.1. Correctness of the algorithm follows from Lemma 4.3.

The running time is dominated by the first step where we compute the value of $M(B, X, C)$ for each choice of parameters. There are $|\Pi|$ choices for B, $|B| + 1 \leq n + 1$ choices for X and at most n choices for C, so there are $O(|\Pi| n^2)$ choices for the parameters. For a particular choice of (B, X, C) computing $M(B, X, C)$ in a naive manner takes time $O(|\Pi| nm)$ and needs $O(|\Pi| n^2)$ table look-ups. Hence the total running time of the algorithm is upper bounded by $O(|\Pi|^2 n^4 m)$.

5 Conclusion

We gave an algorithm with running time $O(n^{12}m)$ for the Weighted Independent Set problem on P_5-free graphs. We did not try to optimize the running time of the algorithm, but it seems difficult to go below $O(n^{10})$ using our approach. Getting an algorithm with a more practically feasible running time would be quite interesting. Our methods seem to break down already on P_6 free graphs, and so a complete classification of the complexity status of Independent Set on graphs with a single connected forbidden induced subgraph remains wide open.

References

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

[2] V. E. Alekseev, *Polynomial algorithm for finding the largest independent sets in graphs without forks*, Discrete Applied Mathematics, 135 (2004), pp. 3–16.

[3] V. E. Alekseev and V. V. Lozin, *Augmenting graphs for independent sets*, Discrete Applied Mathematics, 145 (2004), pp. 3 – 10.

[4] S. Arora and B. Barak, *Computational Complexity - A Modern Approach*, Cambridge University Press, 2009.

[5] B. S. Baker, *Approximation algorithms for np-complete problems on planar graphs*, J. ACM, 41 (1994), pp. 153–180.

[6] E. Balas and C. S. Yu, *On graphs with polynomially solvable maximal-weight clique problem*, Networks, 19 (1989), pp. 247–253.

[7] J. R. S. Blair, P. Heggernes, and J. A. Telle, *A practical algorithm for making filled graphs minimal*, Theor. Comput. Sci., 250 (2001), pp. 125–141.

[8] J. R. S. Blair and B. W. Peyton, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computations, Springer, 1993, pp. 1–30. IMA Volumes in Mathematics and its Applications, Vol. 56.

[9] R. Boliac and V. V. Lozin, *An augmenting graph approach to the stable set problem in P_5-free graphs*, Discrete Applied Mathematics, 131 (2003), pp. 567 – 575.

[10] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: Grouping the minimal separators*, SIAM J. Comput., 31 (2001), pp. 212–232.

[11] ———, *Listing all potential maximal cliques of a graph*, Theor. Comput. Sci., 276 (2002), pp. 17–32.

[12] A. Brandstädt and R. Mosca, *On the structure and stability number of P_5- and co-chair-free graphs*, Discrete Applied Mathematics, 132 (2003), pp. 47 – 65. Stability in Graphs and Related Topics.

[13] A. Brandstädt, J. P. Spinrad, et al., *Graph classes: a survey*, no. 3, Siam, 1999.

[14] H. Broersma, T. Kloks, D. Kratsch, and H. Müller, *Independent sets in asteroidal triple-free graphs*, SIAM J. Discrete Math., 12 (1999), pp. 276–287.

[15] P. Buneman, *A characterization of rigid circuit graphs*, Discrete Math., 9 (1974), pp. 205–212.

[16] B. N. Clark, C. J. Colbourn, and D. S. Johnson, *Unit disk graphs*, Discrete Math., 86 (1990), pp. 165 – 177.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (3. ed.)*, MIT Press, 2009.

[18] D. Corneil, H. Lerchs, and L. Burlingham, *Complement reducible graphs*, Discrete Applied Mathematics, 3 (1981), pp. 163 – 174.

[19] B. Courcelle, J. A. Makowsky, and U. Rotics, *Linear time solvable optimization problems on graphs of bounded clique-width*, Theory Comput. Syst., 33 (2000), pp. 125–150.

[20] G. A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.

[21] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[22] H. Flier, M. Mihalák, A. Schöbel, P. Widmayer, and A. Zych, *Vertex disjoint paths for dispatching in railways*, in Proceedings of ATMOS, 2010, pp. 61–73.

[23] F. V. Fomin and Y. Villanger, *Finding induced subgraphs via minimal triangulations*, in STACS, vol. 5 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 383–394.

[24] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Series of Books in the Mathematical Sciences, W. H. Freeman and Co., 1979.

[25] F. Gavril, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, J. Combin. Theory Ser. B, 16 (1974), pp. 47–56.

[26] M. U. Gerber and V. V. Lozin, *On the stable set problem in special P_5-free graphs*, Discrete Applied Mathematics, 125 (2003), pp. 215–224.

[27] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

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

[29] P. Heggernes, *Minimal triangulations of graphs: A survey*, Discrete Mathematics, 306 (2006), pp. 297–317.

[30] C.-W. Ho and R. Lee, *Counting clique trees and computing perfect elimination schemes in parallel*, Information Processing Letters, 31 (1989), pp. 61 – 68.

[31] C. T. Hoàng, M. Kamiński, V. Lozin, J. Sawada, and X. Shu, *Deciding k-colorability of P_5-free graphs in polynomial time*, Algorithmica, 57 (2010), pp. 74–81.

[32] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.

[33] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, 1972, pp. 85–103.

[34] T. Kloks, D. Kratsch, and J. Spinrad, *On treewidth and minimum fill-in of asteroidal triple-free graphs*, Theor. Comput. Sci., 175 (1997), pp. 309–335.

[35] R. J. Lipton and R. E. Tarjan, *Applications of a planar separator theorem*, SIAM J. Comput., 9 (1980), pp. 615–627.

[36] D. Lokshtanov, D. Marx, and S. Saurabh, *Lower bounds based on the exponential time hypothesis*, Bulletin of the EATCS, 105 (2011), pp. 41–72.

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

[38] V. V. Lozin and R. Mosca, *Independent sets in extensions of $2k_2$-free graphs*, Discrete Applied Mathematics, 146 (2005), pp. 74 – 80.

[39] F. Maffray and G. Morel, *On 3-colorable P_5-free graphs*, SIAM Journal on Discrete Mathematics, 26 (2012), pp. 1682–1708.

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

[41] R. Mosca, *Some observations on maximum weight stable sets in certain P_5-free graphs*, European Journal of Operational Research, 184 (2008), pp. 849 – 859.

[42] R. Mosca, *Some results on stable sets for k-colorable P_6-free graphs and generalizations.*, Discrete Mathematics & Theoretical Computer Science, 14 (2012), pp. 37–56.

[43] A. Parra and P. Scheffler, *Characterizations and algorithmic applications of chordal graph embeddings*, Discrete Applied Mathematics, 79 (1997), pp. 171–188.

[44] S. Poljak, *A note on stable sets and colorings of graphs*, Commentationes Mathematicae Universitatis Carolinae, 15 (1974), pp. 307 – 309.

[45] B. Randerath and I. Schiermeyer, *On maximum independent sets in P_5-free graphs*, Discrete Applied Mathematics, 158 (2010), pp. 1041–1044.

[46] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

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

[48] S. M. Takeshi Fukuda, Yasuhiko Morimoto and T. Tokuyama, *Data mining with optimized two-dimensional association rules*, ACM Transactions on Database Systems, 26 (2001), pp. 179–213.

[49] J. R. Walter, *Representations of rigid cycle graphs*, PhD thesis, Wayne State University, 1972.