University of Warsaw
Faculty of Mathematics, Informatics, and Mechanics
Doctoral School of Exact and Natural Sciences

Jana Masaříková

Computational Complexity of Combinatorial Problems in Hereditary Graph Classes

Doctoral Dissertation

Supervisor: prof. dr hab. Marcin Pilipczuk
Institut of Informatics
University of Warsaw

September 2024

Supervisor's statement

Hereby I confirm that the presented thesis was prepared under my supervision and that it fulfills the requirements for the degree of Doctor in the field of Natural Sciences in the discipline of computer and information sciences

Date ....

prof. dr hab. Marcin Pilipczuk

Author's statement

Hereby I declare that the presented thesis was prepared by me and none of its contents was obtained by means that are against the law.

The thesis has never before been a subject of any procedure for obtaining an academic degree.

Moreover, I declare that the presented version is identical to the attached electronic version.

Date ....

Jana Masaříková

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie stopnia doktora w dziedzinie nauk ścisłych i przyrodniczych w dyscyplinie informatyka

Date . . . .

prof. dr hab. Marcin Pilipczuk

Oświadczenie autora pracy

Oświadczam, że niniejsza rozprawa doktorska została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami. Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem stopnia doktora w innej jednostce. Niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Date . . . .

Jana Masaříková

Abstract

This dissertation thesis is based on a collection of results from structural graph theory and algorithms focusing on the complexity of certain NP-complete graph problems when the input is restricted to a hereditary graph class. A graph class is hereditary if it has the natural property of being closed under the vertex deletion operation, consequently, being characterized by a set of forbidden induced subgraphs. The thesis mainly examines graph classes defined by one or two forbidden induced subgraphs, referred to as H-free or (H_1, H_2)-free graphs, respectively. The common approach to problems described in the thesis involves exploiting the structural properties of a given graph class via the absence of particular induced subgraphs to design more efficient algorithms.

The first part of the thesis focuses on specific graph problems, considering the graph classes that advance the overall understanding of the particular problem. It begins with the Maximum Weight Independent Set problem and the only possible candidate for polynomial time algorithm among H-free graphs: H being a graph whose each connected component is a tree with at most three leaves. A structural result for $S_{t,t,t}$-free graphs is presented utilizing so-called extended strip decomposition, which is well-designed for solving the Maximum Weight Independent Set problem within these graphs. As an application, a subexponential-time algorithm with running time $2^{\mathcal{O}(\sqrt{n}\log n)}$ and a quasi-polynomial time approximation scheme with running time $2^{\mathcal{O}(\varepsilon^{-1}\log^5 n)}$ are presented, improving and simplifying existing algorithms. The second problem considered in the thesis is the 3-Coloring problem. A polynomial-time algorithm for $(2P_4, C_5)$-free graphs is presented. This marks the first attempt to tackle the 3-Coloring problem in $2P_4$-free graphs, besides the challenging case of P_8-free graphs, $2P_4$-free graphs are the only other H-free class, where H has eight vertices, for which a polynomial algorithm is not yet known. Next, the thesis explores vertex deletion problems, specifically focusing on the problem of vertex deletion into bipartite permutation graphs asking whether we can delete k vertices from a general graph to obtain a bipartite permutation graph. This NP-complete problem is shown to be fixed-parameter tractable with an algorithm that runs in $\mathcal{O}(9^k \cdot |V(G)|^9)$.

The second part is tailored to identifying graph classes among (H_1, H_2)-free graphs that possess strong common structural properties. These properties can simplify a broad range of problems without the need for algorithms to be tailored to a specific class or problem. Namely, the thesis focuses on the boundedness of the clique-width in atoms—graphs without a clique cut-set. Multiple graph problems, including Coloring and Maximum Independent Set, that are polynomial-time solvable on atoms of hereditary graph class are so on the entire class. A comprehensive study of the clique-width of atoms of (H_1, H_2)-free graphs for all possible pairs of forbidden graphs H_1, H_2 is initiated, with all but 18 cases classified. A new pair, $(2P_2, \overline{P_2 + P_3})$, is identified where the clique-width of atoms is bounded, while the clique-width of the entire class is unbounded.

Finally, the focus is shifted from algorithms to packings and hittings. In 1981, Tuza conjectured that in a graph without $k+1$ edge-disjoint triangles, it suffices to delete at most $2k$ edges to obtain a triangle-free graph. The thesis confirms the conjecture for threshold graphs and additionally for co-chain graphs with sides of the same size divisible by four.

Keywords: forbidden induced subgraphs, graph coloring, maximum independent set, vertex deletion, clique-width, Tuza's conjecture

Streszczenie

Niniejsza praca doktorska zawiera wyniki ze strukturalnej i algorytmicznej teorii grafów, ze szczególnym uwzględnieniem złożoności wybranych problemów NP-zupełnych dla grafów, gdy graf wejściowy jest ograniczona do pewnej dziedzicznej klasy grafów. Mówimy, że klasa grafów jest dziedziczna, jeśli ma naturalną własność bycia zamkniętą na operację usuwania wierzchołków, co jest równoważne byciu charakteryzowaną przez zbiór zakazanych indukowanych podgrafów. Praca koncentruje się głównie na klasach grafów zdefiniowanych przez jeden lub dwa zakazane podgrafy indukowane, określane odpowiednio jako grafy „bez H" lub „bez (H_1, H_2)". Typowe podejście do problemów opisanych w pracy polega na wykorzystaniu właściwości strukturalnych danej klasy grafów, w szczególności braku określonych indukowanych podgrafów, w celu zaprojektowania bardziej efektywnych algorytmów.

Pierwsza część pracy koncentruje się na indywidualnych problemach grafowych, rozważając klasy grafów, które przyczyniają się do lepszego zrozumienia danego problemu. Zaczynamy od problemu maksymalnego zbioru niezależnego (w wariancie ważonym) i jedynej możliwej kandydatury na istnienie algorytmu działającego w czasie wielomianowym wśród klas grafów bez H: gdy H jest grafem, którego każda spójna składowa jest drzewem mającym co najwyżej trzy liście. Wynik strukturalny dla grafów bez $S_{t,t,t}$ jest przedstawiony z wykorzystaniem rozkładu zwanego tzw. extended strip decomposition, który jest dobrze przystosowany do rozwiązywania problemu maksymalnego zbioru niezależnego w tych klasach grafów. Jako zastosowanie, zaprezentowany zostaje algorytm podwykładniczy o czasie działania $2^{\mathcal{O}(\sqrt{n}\log n)}$ oraz quasi-wielomianowy schemat aproksymacji o czasie działania $2^{\mathcal{O}(\varepsilon^{-1}\log^5 n)}$, co poprawia i upraszcza istniejące wcześniej algorytmy. Drugim problemem rozważanym w pracy jest problem 3-kolorowania. Przedstawiony zostaje algorytm wielomianowy dla grafów bez $(2P_4, C_5)$. Jest to pierwsza próba rozwiązania problemu 3-kolorowania w grafach bez $2P_4$; oprócz trudnego przypadku grafów bez P_8, grafy bez $2P_4$ są jedyną inną klasą grafów bez H, gdzie H ma osiem wierzchołków, dla której nie jest jeszcze znany algorytm wielomianowy. Następnie praca skupia się na problemach usuwania wierzchołków, szczególnie analizując problem redukcji grafu do dwudzielnego grafu permutacji poprzez usuwanie wierzchołków. Zasadniczym pytaniem jest, czy możliwe jest usunięcie k wierzchołków z dowolnego grafu, aby uzyskać dwudzielny graf permutacji. Ten NP-zupełny problem okazuje się mieć algorytm parametryzowany, działający w czasie $\mathcal{O}(9^k \cdot |V(G)|^9)$.

Druga część pracy koncentruje się na wyznaczeniu tych klas grafów, wśród grafów bez (H_1, H_2), które posiadają silne właściwości strukturalne. Właściwości te mogą dawać efektywne algorytmy dla szerokiego zakresu problemów bez konieczności dostosowywania algorytmów do specyficznej klasy lub problemu. Praca skupia się na ograniczoności parametru szerokości klikowej w atomach — grafach bez separatora będącego kliką. Wiele problemów dotyczących grafów, w tym problem kolorowania oraz znajdowania maksymalnego zbioru niezależnego, mają następującą właściwość: jeśli, dla pewnej dziedzicznej klasy grafów, są one rozwiązywalne w czasie wielomianowym na atomach w tej klasie, to są one również rozwiązywalne w całej klasie. Prezentujemy przekrojowe badanie szerokości klikowej w atomach grafów bez (H_1, H_2) dla wszystkich możliwych par zakazanych grafów H_1, H_2, z wyjątkiem 18 przypadków. W szczególności, dla jednej nowej pary, $(2P_2, \overline{P_2 + P_3})$, pokazujemy, że szerokość klikowa w atomach jest ograniczona, podczas gdy szerokość klikowa w całej klasie jest nieograniczona.

Na koniec przenosimy się do zagadnień pakowania i pokrywania. W 1981 roku Tuza wysunął hipotezę, że w grafie bez $k+1$ trójkątów rozłącznych krawędziowo wystarczy usunąć co najwyżej $2k$ krawędzi, aby uzyskać graf bez trójkąta. Potwierdzamy tę hipotezę dla grafów progowych, a także dla grafów ko-łańcuchowych, gdzie strony mają tę samą liczbę wierzchołków podzielną przez cztery.

Słowa kluczowe: zakazane podgrafy indukowane, kolorowanie grafów, maksymalny zbiór niezależny, problemy usuwania wierzchołków, klikowa szerokość, hipoteza Tuzy

Acknowledgements. First and foremost, I would like to express my deepest gratitude to my supervisor, Marcin Pilipczuk, for his guidance, support, and understanding of my decisions. Working with him has been an incredible learning experience for me.

I would also like to thank my wonderful office colleagues: Lars Jaffke, Paloma de Lima, Karolina Okrasa, and Paweł Rzążewski. I am truly grateful that we became more than just coworkers, sharing countless moments of joy and laughter. A special thanks to Karolina who was always there whenever I needed anything. Warm thanks to Marthe Bonamy, for her smile and care during her visit to Warsaw at the start of my studies, and to Łukasz Bożyk, for always assisting with formalities. I extend my gratitude to Daniël Paulusma, who introduced me to this research topic and provided me with valuable feedback in the early stages of my career. Finally, I thank my former colleagues from Prague, particularly Jiří Fiala.

I am grateful to my family, who, despite the initial shock of my decision to move abroad, eventually accepted and supported me with love and understanding.

Finally, heartfelt thanks to my beloved husband, Tomáš, who supported my decision to pursue a PhD, even though this journey made things more complicated for him. To our daughter, Amálka, who joined us on many conference trips, showing endless patience and positive energy as she stayed by my side while I worked.

Contents

1 Introduction

Many hard graph problems become tractable when restricting the input to certain graph classes. This raises two central questions: “for which graph classes does a graph problem become tractable” and “for which graph classes does it stay computationally hard?”

What do we mean by *tractable*? The complexity of many graph problems varies based on the restriction imposed on the input graph. These problems can range from being solvable in polynomial time, admitting quasipolynomial-time or subexponential-time algorithms, to being NP-hard and hard to approximate. A methodological study of this behavior leads to the following more specific question:

> For which structures, does their absence from the input graph make a graph problem easier, and to what extent?

The “absence of structures” notion can be made precise by specifying the forbidden structure and the containment relation, such as a minor, topological minor, induced minor, subgraph, or induced subgraph. The last one — induced subgraph relation — is the weakest one and therefore the most expressible, particularly when multiple forbidden structures are considered. This leads us to study graph problems in various *hereditary* graph classes, that is, graph classes closed under vertex deletion. Besides capturing many well-known classes, the framework of hereditary graph classes also enables us to perform a *systematic* study of graph problems. This is because every hereditary graph class $\mathcal{G}$ can be uniquely characterized by a minimal (though not necessarily finite) set $\mathcal{F}_{\mathcal{G}}$ of forbidden induced subgraphs. While a general classification of *all* hereditary graph classes concerning the complexity of classic graph problems may be too complex, classifying graph classes with one (or a few) forbidden induced subgraphs appears more feasible. Therefore, this becomes our primary focus.

If $\mathcal{F}_{\mathcal{G}} = \{H\}$ or $\mathcal{F}_{\mathcal{G}} = \{H_1, H_2\}$, then $\mathcal{G}$ is said to be H-free or (H_1, H_2)-free, respectively. These graph classes already exhibit a rich structure, and studying their properties has led to deep insights into solving graph problems. Surveys on graph problems or parameters specifically restricted to (H_1, H_2)-free graph classes provide evidence of this [71, 104]. Additionally, the complexity of a given problem in H-free graphs may indicate the impact of forbidding H as an induced subgraph on the complexity of the graph problem in more general settings.

The selection of suitable forbidden subgraphs depends on the specific graph problem under consideration. Firstly, we turn our attention to two fundamental problems in graph theory: graph coloring and maximum independent set. These problems are cornerstones of combinatorial optimization and have wide-ranging applications across various domains, including scheduling, resource allocation, and network design. Secondly, we shift our focus from addressing each problem individually to an approach aimed at answering complexity questions for a broad set of problems simultaneously. This direction involves studying graph width parameters—such as treewidth, clique-width, and others—which have proven effective in making such results possible when restricting on graphs with forbidden induced subgraphs [71, 110, 120, 130, 171]. This approach can be seen as “stronger” than tackling individual problems separately, as it has the potential to resolve the complexity questions for numerous problems in one go thanks to meta-theorems when the width is bounded.

However, it can also be viewed as “weaker” because a “negative” result——unbounded width——does not ruin the tractability results. Consider for instance the class of split graphs where the clique-width is unbounded but the maximum independent set is trivially determined, thus polynomial-time solvable.

1.1 Maximum Independent Set

The Maximum Weight Independent Set (MWIS) problem is a fundamental problem in combinatorial optimization, where the objective is to find an independent set of vertices in a graph such that the sum of their weights is maximized.

Maximum Weight Independent Set	
Instance:	Graph G with nonnegative vertex weights.
Question:	What is the maximum total weight of an independent set of G?

Some graph classes allow polynomial-time solutions of the MWIS problem (e.g., chordal graphs), while in general graphs the problem is NP-hard and hard to approximate within $n^{1-\varepsilon}$ factor [173, 113]. Moreover, in general graphs MWIS cannot even be solved in subexponential time unless the Exponential-Time Hypothesis (ETH) fails. The problem of MWIS remains NP-hard in H-free graphs for most graphs H as observed by Alekseev in 1980s [7, 8]. We delve deeper into this observation here: First, MWIS is NP-hard in subcubic graphs [95], which are H-free whenever H contains a vertex of degree at least four. Second, by subdividing a single edge twice, the size of a maximum independent set increases by one. This well-controlled operation allows the construction of an auxiliary graph that is H-free whenever H contains a cycle or two vertices of degree three in the same connected component. Consequently, MWIS remains NP-hard unless every connected component of H is either a path or a subdivided claw (three paths connected at a common vertex). Thus, significant attention has been devoted to P_t-free graphs (graphs excluding a path on t vertices) and to $S_{t,t,t}$-free graphs (a tree with three leaves connected to the root by paths of length t.)

Since the 1980s, it has been known that MWIS is polynomial-time solvable on P_4-free graphs, also known as cographs, due to their strong structural properties causing many problems to admit a polynomial-time algorithm there (see, e.g. [62]). Similarly, MWIS is polynomial time solvable in $S_{1,1,1}$-free graphs [155, 164], commonly referred to as claw-free graphs. Claw-free graphs are closely related to line graphs where finding a maximum-weight independent set corresponds to a maximum-weight matching in the original graph. In claw-free graphs, the use of augmenting paths allows for the efficient determination of a maximum-weight independent set. The next smallest subdivided claw is $S_{1,1,2}$, also known as the fork, for which a polynomial algorithm has been shown by Alekseev [9] in 2004. However, progressing beyond this point becomes significantly more challenging. Currently, $S_{1,2,2}$ represents the smallest open case where the polynomiality of MWIS has not been confirmed yet.

For many years, progress was limited to partial results within subclasses only until the area started to develop rapidly around 2014. In 2014, Lokshtanov, Vatshelle, and Villanger [145] showed that MWIS is solvable in a polynomial time for P_5-free graphs using a framework of potential maximal cliques. The result was later generalized to

P_6-free graphs by Grzesik, Klimošová, Pilipczuk, and Pilipczuk in 2019 [109]. However, the complexity of MWIS for P_7-free graphs remains open. Several related partial results have been obtained, for instance [3], where authors showed that certain classes[1] admit a polynomial number of minimal separators and, consequently, allow for a polynomial algorithm for MWIS [91, 23]. For follow-up works in this direction, refer to Section 1.8. Another notable result includes a polynomial algorithm for $C_{>4}$-free graphs [4].

Another direction involves simplifying the problem by imposing additional input restrictions. Abrishami, Chudnovsky, Dibek, and Rzążewski announced a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs of bounded degree [2]. It was later improved in [5] where the same set of authors along with Pilipczuk provided a polynomial-time algorithm for MWIS in graphs excluding a fixed graph whose every component is a subdivided claw as an induced subgraph, and a fixed biclique as a subgraph.

Switching back to the main direction but with a focus on relaxing the aim to get a polynomial algorithm, we highlight some cornerstone results. Around 2018, Bacsó, Lokshtanov, Marx, Pilipczuk, Tuza, and van Leeuwen [11] observed that the classic Gyárfás' path argument, originally developed to show that for every fixed t the class of P_t-free graphs is χ-bounded [111, 112], also yields a subexponential-time algorithm for MWIS in P_t-free graphs. The crucial corollary of the Gyárfás' path argument is encapsulated in the following theorem:

Theorem 1 (Gyárfás' path)**.** *Given an n-vertex graph G with non-negative vertex weights, one can in polynomial time find an induced path Q in G such that every connected component of $G - N[V(Q)]$ has weight at most half of the total weight of $V(G)$.*

For P_t-free graphs the said path Q has at most $t-1$ vertices. Bacsó et al. observed that branching either on the highest degree vertex (if its degree is larger than $\sqrt{n}$) or on the entire set $N[V(Q)]$ (otherwise) results in the algorithm with running time bound exponential in $\sqrt{n} \cdot \mathrm{poly}(t, \log n)$.

Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [53] added an observation that a simple branching algorithm is able to get rid of *heavy* vertices, that is, vertices of the input graph whose neighborhood contains a large fraction of the sought independent set. Once this branching is executed and the graph does not have heavy vertices, the set $N[Q]$ (from Theorem 1) contains only a small fraction of the sought solution and, if one aims for an approximation algorithm, can be just sacrificed, yielding a quasipolynomial-time approximation scheme (QPTAS) for MWIS in P_t-free graphs. Using this as a starting point and leveraging on the celebrated *three-in-a-tree* theorem of Chudnovsky and Seymour [56], they developed a much more sophisticated QPTAS and a subexponential algorithm (with running time bound $2^{n^{8/9}\mathrm{poly}(\log n, t)}$) for MWIS in $S_{t,t,t}$-free graphs in 2020.

In 2020, Gartland and Lokshtanov [97] brought a breakthrough result in P_t-free graphs. Consider a natural branching algorithm for MWIS: in each connected component independently pick a vertex (*pivot*) v and branch into two cases: either v is in the sought independent set, thus recursing on $G - N[v]$ (successful branch), or not, recursing on $G - v$. The performance of such an algorithm highly depends on the choice of the pivot

[1]They showed that the class of (theta, pyramid, prism, turtle)-free graphs, implying also a more known class of (even-hole,pyramid)-free graphs, admits polynomial many minimal separators.

v. Gartland and Lokshtanov showed how to properly choose the pivot in P_t-free graphs to guarantee large progress in the successful branch, leading to a quasipolynomial-time running bound for MWIS in P_t-free graphs.

Inspired by this work, Pilipczuk, Pilipczuk, and Rzążewski [159] provided an arguably simpler measure that led to an improved, yet still quasipolynomial, running-time bound. In both approaches, Gyárfás' path played a significant role as certain vertices within this path proved to be excellent candidates for the pivot. In the latter, the authors analyzed the space of all induced paths (observe that it is bounded by $\mathcal{O}(n^{t-1})$ in P_t-free graphs) to guide the branching algorithm. Specifically, they categorized induced paths into "buckets" based on their endpoints, where each bucket $B_{u,v}$ contains all induced paths with endpoints u and v. By considering these buckets and selecting Q from Theorem 1, they ensured that $N[Q]$ intersects all paths from at least half of the buckets. Given that $|Q| \leq t$, a vertex whose neighborhood intersects at least $1/t$ paths from at least $1/2t$ buckets can always be found and used as a pivot. Thus, after a logarithmic number of branchings on such vertices, the graph gets disconnected into multiplicatively smaller components.

These developments have been subsequently generalized to a larger class of problems beyond MWIS and to $C_{>t}$-free graphs (graphs without induced cycles of length more than t) [100].

In 2022, we extended this line of research and showed how to use the three-in-a-tree theorem to obtain an important structural property of $S_{t,t,t}$-free graphs. This property serves as an analog to Gyárfás' path argument, thereby facilitating further progress in analyzing $S_{t,t,t}$-free graphs. We discuss the result in greater detail later. Using the new structural insight, we provided a simpler subexponential algorithm with a better running time $2^{\mathcal{O}(\sqrt{n}\log n)}$ and a QPTAS with running time $2^{\mathcal{O}(\varepsilon^{-1}\log^5 n)}$. Both results improved and simplified the previously known algorithms by Chudnovsky et al. [53]. Moreover, the structural result found other applications, such as in [5]. Most importantly, it contributed to the second breakthrough in the field: the development of a quasipolynomial algorithm for $S_{t,t,t}$-free graphs, presented in 2023 by Gartland, Lokshtanov, Masařík, Pilipczuk, Pilipczuk, and Rzążewski [99]. The authors build upon a branching similar to the one used in the first quasipolynomial algorithm for P_t-free graphs[97] combined with our structural result and more involved work with balanced separators. This progress corroborates the conjecture that MWIS is polynomial-time solvable in H-free graphs for all the open cases left by the original hardness reductions, that is, whenever H is a forest whose every connected component has at most three leaves.

Our focus. We focus on $S_{t,t,t}$-free graphs. Our main contribution lies in providing a structural result of $S_{t,t,t}$-free graphs that became an analog of the Gyárfás' path argument. The result is covered by the paper [150]. We build upon the concept of an *extended strip decomposition*, which was developed by Chudnovsky and Seymour in their project on understanding claw-free graphs [55]. For a formal definition of extended strip decomposition, we refer to Section 3.1. For an intuition to develop, we remark that in extended strip decomposition of a graph, we can distinguish *particles* being induced subgraphs of the graph. The crucial property in the context of solving MWIS is that we can recurse on individual particles, compute the maximum weight independent sets within them, and combine the results into a maximum weight independent set for the entire graph by employing a maximum weight matching algorithm on an auxiliary graph (cf. [53]).

Thus, an extended strip decomposition of a graph with particles of multiplicatively smaller size is very useful for recursion. It functions similarly as splitting the graph into connected components of multiplicatively smaller size, as it is in the case of the components of $G - N[V(Q)]$ in Theorem 1.

With the above discussion in mind, we can now state our main technical result.

Theorem 2. *Given an n-vertex graph G with nonnegative vertex weights and integer* $t \geq 1$*, one can in polynomial time either:*

- *output an induced copy of* $S_{t,t,t}$ *in G, or*
- *output a set* $\mathcal{P}$ *consisting of at most* $11 \log n + 8$ *induced paths in G, each of length at most* $t+1$ *and a rigid* extended strip decomposition *of* $G - N[\bigcup_{P \in \mathcal{P}} V(P)]$ *whose every particle has weight at most half of the total weight of* $V(G)$.

The proof combines the Gyárfás' path approach with the three-in-a-tree theorem, that for a given set Z of at least three vertices outputs in a polynomial time (originally in $\mathcal{O}(n^4)$ [56], improved to $\mathcal{O}(n)$ [140]) either an extended strip decomposition of G with respect to Z (vertices from Z have special positions in the decomposition, we omit the details here) or an induced subtree of G connecting at least three vertices of Z.

We start with a Gyárfás' path Q navigating towards the component of the largest weight in G. If the path is *short* ($Q \leq 3t+1$), we immediately get the desired outcome (by constructing a *trivial* extended strip decomposition of $G - N[Q]$ where each connected component is put into a separate vertex of the extended strip decomposition; in this situation, each particle consists of one connected component of $G \setminus N[Q]$). Otherwise, we proceed by splitting Q into thirds, removing the neighborhoods (outside Q) of the first t vertices of each part, and applying the three-in-a-tree theorem to the starting vertices of each of these three segments. This ensures that the first output of the three-in-a-tree theorem is an induced $S_{t,t,t}$. In the output of the extended strip decomposition, we either already have the desired output or we continue by recursing on the particle of weight greater than half of the total weight of G. The properties of the extended strip decomposition ensure that at most two segments of the split path Q can touch it. These segments are then used in place of Q as the new induced paths for the subsequent step of the recursion. We continue this process, splitting the larger of the two segments as before, which significantly reduces the number of vertices in Q. We note that we will always have at most two starting induced paths for the remaining recursion steps and we will always ask for a new extended strip decomposition. As we measure the progress by the number of the vertices of our original Q and stop when there are at most $3t+1$ of them, the recursion has $11 \log n$ steps. Apart from the technical details, this is where the factors from the second output of Theorem 2 came from.

Combining Theorem 2 with previously known algorithmic techniques, we derive two algorithms for MWIS in $S_{t,t,t}$-free graphs. Actually, our algorithms work in a slightly more general setting, for $sS_{t,t,t}$-free graphs (the forbidden induced graph contains s connected components, each isomorphic to $S_{t,t,t}$). Recall that by the observation of Alekseev [7, 8] the only graphs H, for which we can hope for tractability results for MWIS in H-free graphs, are forests whose every component has at most three leaves. We observe that each such H is contained in $sS_{t,t,t}$, for some s and t depending on H. Thus algorithms for $sS_{t,t,t}$-free graphs, for every s and t, cover *all* potential positive cases.

First, we observe that the statement of Theorem 2 seamlessly combines with the method how [11] obtained a subexponential-time algorithm for MWIS in P_t-free graphs. As a result, we obtain a subexponential-time algorithm for MWIS in $sS_{t,t,t}$-free graphs with improved running time as compared to [53].

Theorem 3. *Given an n-vertex graph G with weights on vertices and integers* $s, t \geq 1$*, one can in time exponential in* $\mathcal{O}(\sqrt{stn}\log n)$ *output one of the following outcomes:*
1. an induced $sS_{t,t,t}$ *in G, or*
2. an independent set in G of maximum possible weight.

Second, we observe that the statement of Theorem 2 again seamlessly combines with the method how [53] obtained a QPTAS for MWIS in P_t-free graphs, obtaining an arguably simpler QPTAS for MWIS in $sS_{t,t,t}$-free graphs with an improved running time (compared to [53]).

Theorem 4. *Given an n-vertex graph G with weights on vertices, integers* $s, t \geq 1$*, and a real* $\varepsilon > 0$*, one can in time exponential in* $\mathcal{O}(\varepsilon^{-1} st \log^5 n)$ *output one of the following outcomes:*
1. an induced $sS_{t,t,t}$ *in G, or*
2. an independent set in G that is within a factor of $(1-\varepsilon)$ *of the maximum possible weight.*

The discussed result has already found several applications. As mentioned, most notably, it has become a crucial component, used as a black box, in the quasipolynomial algorithm for MWIS in $S_{t,t,t}$-free graphs [99]. The key structural ingredient for the quasipolynomial algorithm lies in the following theorem: For a fixed integer $t \geq 1$, 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 $\mathcal{O}(\log |V(G)|)$ vertex neighborhoods in G, or an extended strip decomposition of G with each particle of weight multiplicatively smaller than the weight of G. We remark that the third output provides an extended strip decomposition of the entire graph, in contrast to Theorem 2.

We conclude with a recent improvement of Theorem 2 in [24]. Together with Bourneuf, Nadara, and Pilipczuk, we eliminated the logarithmic factor in the second output of Theorem 2. Namely, we showed that removing neighborhoods of a constant number (in contrast to logarithmic) of short paths (specifically, in total, the neighborhood of $3t + \mathcal{O}(1)$ vertices) is sufficient to obtain an extended strip decomposition with each particle of weight at most half of the total weight of our graph. This improvement shows the strongest possible bound up to a constant. Moreover, it significantly simplifies some of the existing results, for instance, the polynomial-time algorithm for $sS_{t,t,t}$-free graphs that additionally exclude fixed biclique as a subgraph [5]. We discuss the result in more detail in Section 3.4, Conclusion of Chapter 3.

1.2 Coloring

Graph coloring is a notoriously known and well-studied concept in both graph theory and theoretical computer science. A *k-coloring* of a graph $G = (V, E)$ is defined as a mapping $c : V \to \{1, \ldots, k\}$ which is *proper*, i.e., it assigns distinct colors to adjacent vertices. The k-Coloring problem asks whether a given graph admits a k-coloring.

k-Coloring *Instance:* Graph G *Question:* Does G admits a proper coloring that uses at most k colors?

For any $k \geq 3$, the k-Coloring is a well-known NP-complete problem [131] in general graphs. We also define a more general *list-k-coloring* where each vertex v has a list $P(v)$ of admissible colors such that $P(v) \subseteq \{1, \ldots, k\}$. In that case, the coloring function c, in addition to being proper, has to respect the lists, that is, $c(v) \in P(v)$ for every vertex v. We call the function P, a *k-list assignment*. If k is fixed, then we obtain the following generalization of the k-Coloring problem:

List-k-Coloring *Instance:* Graph G, a k-list assignment P *Question:* Does there exist a proper coloring of G that respects P?

In recent years, a lot of attention has been paid to determining the complexity of k-Coloring of H-free graphs. Classical results imply that for every $k \geq 3$, k-Coloring of H-free graphs is NP-complete if H contains a cycle [84] or an induced claw [124, 143]. Hence, it remains to consider the cases where H is a *linear forest*, i.e., a disjoint union of paths. Focusing on H being connected, that is $H = P_t$ for some t, the situation around the complexity of (List) k-Coloring for $k \geq 4$ has been resolved completely. The (List) k-Coloring on P_t-free graphs is NP-complete [125] if $k = 4, t \geq 7$ or $k \geq 5, t \geq 6$, while the cases for $k \geq 1, t = 5$ are polynomial-time solvable [121]. In fact, k-coloring is polynomial-time solvable on $sP_1 + P_5$-free graphs for any $s \geq 0$ [66].

An interesting difference emerges between 4-Coloring and List-4-Coloring in the case where $k = 4, t = 6$. 4-Coloring (even the precoloring extension problem with four colors) in P_6-free graphs is polynomial-time solvable [167, 58, 59] while List-4-Coloring is NP-complete [105].

Shifting our focus to the complexity of 3-Coloring, we find that it is less well understood, despite the amount of research interest it received in the past years. Significant progress has been made in 2020 when Pilipczuk, Pilipczuk, and Rzążewski [158] presented a quasipolynomial algorithm for 3-Coloring in P_t-free graphs running in time $n^{O(\log^2(n))}$ on n-vertex P_t-free graphs (t is a constant), extending and simplifying the breakthrough of Gartland and Lokshtanov [97] for MWIS. Thanks to this result, it is widely believed that 3-Coloring is polynomial-time solvable in P_t-free graphs for any fixed t (and, thus, in any H-free graph left by the initial NP-hardnesses). In terms of known polynomial-time algorithms, it is well-known that k-Coloring is solvable in linear time on P_4-free graphs (see, e.g., [62]). Bonomo et al. [20] found a polynomial-time algorithm for P_7-free graphs. It remains open to find a polynomial time algorithm for P_8-free graphs. Klimošová et al. [134] completed the classification of 3-Coloring of H-free graphs, for any H on up to 7 vertices. These results were subsequently extended to $P_6 + rP_3$-free graphs, for any $r \geq 0$ [47]. There are still two unresolved cases among graphs with at most eight vertices, namely P_8 and $2P_4$, for which a polynomial algorithm for 3-Coloring has yet to be discovered.

Our focus. We focus on one of the remaining open problems mentioned above, which considers 3-Coloring $2P_4$-free graphs. We impose an additional restriction on the input graph and present a polynomial-time algorithm for 3-Coloring of $(2P_4, C_5)$-free graphs. To the best of our knowledge, this is the first attempt to attack the 3-Coloring of $2P_4$-free graphs.

Theorem 5. *3-Coloring is polynomial-time solvable on $(2P_4, C_5)$-free graphs.*

Grötschel, Lovász, and Schrijver [107] showed that the k-Coloring problem on *perfect graphs* can be solved in polynomial time. According to the Strong Perfect Graph Theorem [54], a graph is perfect if and only if it contains neither an odd-length induced cycle nor the complement of an odd-length induced cycle with at least five vertices. Given that K_4 and $\overline{C_7}$ graphs are not 3-colorable, we can assume that our graph is $(2P_4, C_5, \overline{C_7}, K_4)$-free. Moreover, since $K_4 \subseteq \overline{C_\ell}$ for $\ell \geq 8$ and $2P_4 \subseteq C_\ell$ for $\ell \geq 10$, our graph is either perfect or it contains an induced C_7 or C_9. In the former case, we apply the aforementioned polynomial-time algorithm for perfect graphs. For the latter cases, we further analyze the graph in two subcases that cover all the possibilities: (i) $(2P_4, C_5, C_7, \overline{C_7}, K_4)$-free graphs containing an induced C_9, (ii) $(2P_4, C_5, C_9, \overline{C_7}, K_4)$-free graphs containing an induced C_7,

Both cases share the initial steps and overall approach. We start with the particular induced cycle, all possible 3-colorings of it, and then approach the problem as an instance of List-3-Coloring instead. Our general plan is as follows. We analyze the graph concerning the distance to the colored cycle and apply several reductions of lists as one of our base techniques. After several branching steps with a polynomial number of branches in total and suitable structural reductions to the original graph, the algorithm will transform an input instance to a set of polynomially many heavily structured list-3-coloring instances. These structured instances can then be encoded by a 2-SAT formula, whose satisfiability is solvable in linear time [139].

In case (i), only basic reductions are enough to conclude the algorithm. The difficulty lies in the case (ii). We will exploit the fact that once we find an induced P_4, the vertices that are not adjacent to it must induce a P_4-free graph (also known as *cograph*). Cographs, being one of the earliest studied H-free graphs, possess favorable properties, such as that any greedy coloring gives a proper coloring using the least number of colors [46]. We utilize an even stronger result: List-3-Coloring problem on P_4-free graphs can be solved in polynomial time [104].

We note that the 3-coloring algorithm that we develop to prove Theorem 5 cannot be directly extended to solve the more general List-3-Coloring since it uses the 3-coloring algorithm for perfect graphs to deal with graphs avoiding C_7 and C_9. However, apart from this one case, the algorithm works with the more general setting of list-3-coloring.

1.3 Clique-width and Atoms

The main purpose of this part is to answer the questions about which graph classes enable a graph problem to become tractable for a large set of problems simultaneously instead of considering individual problems one by one. Graph width parameters help to make such results possible as in a certain sense, they measure the structural complexity of a graph. Some of the most common width parameters include treewidth, pathwidth, and clique-width. For instance, treewidth or pathwidth measures how "close" the graph is to a tree or a path, respectively. We consider a graph class having *bounded* width if there is a constant c such that the width of all its members is at most c. There are several meta-theorems that provide sufficient conditions for a problem to be tractable on a graph class of bounded width. In our work, we concentrate on clique-width. This parameter is defined via a graph construction process where vertex labels are assigned to vertices during the construction with the idea that once the two vertices share the same label, they must be treated uniformly in subsequent steps. In particular, one can use the following four operations: the 1) creation of a new vertex with label i, 2) the vertex-disjoint union of already constructed labeled graphs, 3) the insertion of all possible edges between vertices of specified labels, and 4) the uniform relabeling of vertices. The clique-width ($\mathrm{cw}(G)$) of a graph G is defined as the smallest number of labels needed to construct G by these four operations.

Courcelle, Makowsky, and Rotics [65] showed that every graph problem definable in the logic MSO_1 is linear-time solvable on graphs of bounded clique-width. Recall that in MSO_1, we can quantify over vertices and sets of vertices, and check their adjacency (see [64] for details on MSO_1). These captures k-Coloring among others. Since then, several clique-width meta-theorems for graph problems not definable in MSO_1 have been developed [86, 102, 135, 162]. The above meta-theorems require a constant-width decomposition of the graph to be part of the input. While it is NP-hard to compute the optimal decomposition for clique-width [88], we can compute a constant-width decomposition in polynomial time for the clique-width thanks to approximation algorithms [157, 90]. In particular, Fomin and Korhonen showed a $(2^{2\,\mathrm{cw}(G)+1}-1)$-approximation algorithm running in quadratic time [90]. There is a long-standing study on boundedness of clique-width for hereditary graph classes (see, for example, [14, 16, 29, 30, 33, 35, 36, 37, 68, 70, 72, 73, 76, 77, 110, 130, 152]). Moreover, clique-width of (H_1, H_2)-free graph classes has been intensively studied, with only five cases left open [71].

Our focus. If a graph class has unbounded clique-width, it does not mean that a graph problem must be NP-hard on this class. Consider for instance Coloring which is polynomial-time solvable on the class of (C_4, P_6)-free graphs [101]. This class contains the class of split graphs and thus has unbounded clique-width [152]. However, our aim is still to use clique-width to solve multiple problems at once. Thus, we need additional tools. In the case of (C_4, P_6)-free graphs, it turned out that the *atoms* (graphs with no clique cut-set) of (C_4, P_6)-free graphs *do* have bounded clique-width. This immediately gives us an algorithm for the whole class of (C_4, P_6)-free graphs due to Tarjan's decomposition theorem [168]. In fact, Tarjan's result holds not only for Coloring; i.e., several of the problems that are polynomial-time solvable on graphs of bounded clique-width are polynomial-time solvable on a hereditary graph class $\mathcal{G}$ if they are so on the *atoms*. Among these problems belong Maximum Clique, MWIS [168] (see [8] for the unweighted variant) and Maximum Induced Matching [38]. The class of (C_4, P_6)-free graphs is not the only known class with this property, consider for instance chordal graphs (i.e., $(C_4, C_5, \dots)$-free graphs). Chordal graphs have unbounded clique-width [152], but chordal atoms are complete graphs [80] and have clique-width at most 2. We refer to [89, 92] for some examples of polynomial-time algorithms for Coloring on hereditary graph classes that exploit the fact that atoms of subclasses of these graph classes have bounded clique-width. Motivated by these examples, we aim to investigate, in a systematic way,

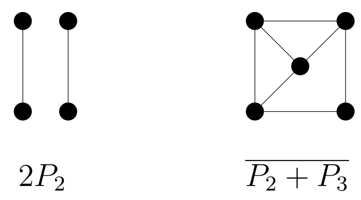


Figure 1.1 The two forbidden induced subgraphs from Theorem 6.

the following natural research question:

> Which hereditary graph classes of *unbounded* clique-width have the property that their atoms have *bounded* clique-width?

For H-free graphs, the restriction to atoms does not yield any algorithmic advantages, as shown by Gaspers et al. [101]; namely a class of H-free graphs has bounded clique-width if and only if its atoms have bounded clique-width (and that is if and only if H is an induced subgraph of P_4). This is no longer true for (H_1, H_2)-free graphs (as evidenced by (C_4, P_6)-free graphs). Hence, we systematically explore the boundedness of clique-width of (H_1, H_2)-free atoms. We prove the existence of another such pair, $(2P_2, \overline{P_2 + P_3})$ (see Figure 1.1) and systematically classify the boundedness of clique-width on (H_1, H_2)-free atoms for all but 18 cases. While it is useful to gain knowledge also about unbounded cases, the positive case has its importance as it gives us new algorithmic results. We state it separately.

Theorem 6. *The class of* $(2P_2, \overline{P_2 + P_3})$*-free atoms has bounded clique-width (whereas the class of* $(2P_2, \overline{P_2 + P_3})$*-free graphs has unbounded clique-width).*

In the proof of Theorem 6, we explore the structure of $(2P_2, \overline{P_2 + P_3})$-free atoms. We split the proof into three separate cases, namely: (i) $(2P_2, \overline{P_2 + P_3})$-free atoms that contain an induced cycle on five vertices; (ii) $(2P_2, \overline{P_2 + P_3}, C_5)$-free atoms that contain an induced cycle on four vertices; (iii) $(2P_2, \overline{P_2 + P_3}, C_4, C_5)$-free atoms. This standard approach enables us to use additional structural properties and, thus, splits the difficulty of the proof into more but easier cases. The case (iii) is known to have clique-width at most two (atoms of split graphs are complete graphs). In the first two cases, we start by fixing an induced cycle on four or five vertices, respectively, and analyze its first and second neighborhoods. We reduce our instance until we obtain a structure that has clearly bounded clique-width. For an example of such a situation where we conclude a bounded clique-width, imagine the following example. Our instance has been reduced to only a constant number of special vertices together with a constant number of independent sets and cliques that are either complete or anti-complete to each other. This graph has clearly a constant clique-width. To highlight one of the places where the property of atoms (i.e., having no clique cut-sets) has helped: Add one additional independent set A to a situation described above such that all its neighbors belong only to one particular clique C, but the connections between A and C might be arbitrary. Recall, that spit graphs have unbounded clique-width, and those are exactly graphs where the vertex set can be partitioned into an independent set and a clique with arbitrary connections in between.

Observe that in the described situation, C is a clique cut-set, thus, A has to be empty; which is exactly the situation in the proof of case (i).

We discuss the proof details and allowed operations to reduce the instance in more detail in Section 5.3. Here we only mention a complication that makes proving the boundedness of clique-width of atoms more difficult. While a class of (H_1, H_2)-free graphs has bounded clique-width if only if the class of $(\overline{H_1}, \overline{H_2})$-free graphs has bounded clique-width, the equivalence no longer holds for classes of (H_1, H_2)-free atoms. For example, (C_4, P_5)-free (and even (C_4, P_6)-free) atoms have bounded clique-width [101], but we will prove that $(\overline{C_4}, \overline{P_5})$-free atoms have unbounded clique-width. Thus, when working with atoms, we need to be careful with performing the complementation operations.

The following theorem[2] captures the boundedness of clique-width of all (H_1, H_2)-free atoms except 18 cases. The remaining cases are stated in Open Problem 2 in Chapter 5. Here we only remark that nine of them have unbounded clique-width when the whole class is considered, the rest of the open cases originated in the five cases where even the boundedness of clique-width of the entire class was not known by the time of writing the paper (and complementary operations); see Open Problem 1 in Chapter 5.

Theorem 7. *For graphs H_1 and H_2, let $\mathcal{G}$ be the class of (H_1, H_2)-free graphs.*

1. *The class of atoms in $\mathcal{G}$ has bounded clique-width if*
   - *(i)* H_1 *or* $H_2 \subseteq_i P_4$
   - *(ii)* $H_1 = \text{paw}$ *or* K_s *and* $H_2 = P_1 + P_3$ *or* tP_1 *for some* $s, t \geq 1$
   - *(iii)* $H_1 \subseteq_i \text{paw}$ *and* $H_2 \subseteq_i K_{1,3} + 3P_1,\ K_{1,3} + P_2,\ P_1 + P_2 + P_3,\ P_1 + P_5,\ P_1 + S_{1,1,2},\ P_2 + P_4,\ P_6,\ S_{1,1,3}$ *or* $S_{1,2,2}$
   - *(iv)* $H_1 \subseteq_i P_1 + P_3$ *and* $H_2 \subseteq_i \overline{K_{1,3} + 3P_1},\ \overline{K_{1,3} + P_2},\ \overline{P_1 + P_2 + P_3},\ \overline{P_1 + P_5},\ \overline{P_1 + S_{1,1,2}},\ \overline{P_2 + P_4},\ \overline{P_6},\ \overline{S_{1,1,3}}$ *or* $\overline{S_{1,2,2}}$
   - *(v)* $H_1 \subseteq_i \text{diamond}$ *and* $H_2 \subseteq_i P_1 + 2P_2,\ 3P_1 + P_2$ *or* $P_2 + P_3$
   - *(vi)* $H_1 \subseteq_i 2P_1 + P_2$ *and* $H_2 \subseteq_i \overline{P_1 + 2P_2},\ \overline{3P_1 + P_2}$ *or* $\overline{P_2 + P_3}$
   - *(vii)* $H_1 \subseteq_i \text{gem}$ *and* $H_2 \subseteq_i P_1 + P_4$ *or* P_5
   - *(viii)* $H_1 \subseteq_i P_1 + P_4$ *and* $H_2 \subseteq_i \overline{P_5}$
   - *(ix)* $H_1 \subseteq_i K_3 + P_1$ *and* $H_2 \subseteq_i K_{1,3}$,
   - *(x)* $H_1 \subseteq_i \overline{2P_1 + P_3}$ *and* $H_2 \subseteq_i 2P_1 + P_3$
   - *(xi)* $H_1 \subseteq_i P_6$ *and* $H_2 \subseteq_i C_4$, *or*
   - *(xii)* $H_1 \subseteq_i 2P_2$ *and* $H_2 \subseteq_i \overline{P_2 + P_3}$.

2. *The class of atoms in $\mathcal{G}$ has unbounded clique-width if*
   - *(i)* $H_1 \notin \mathcal{S}$ *and* $H_2 \notin \mathcal{S}$
   - *(ii)* $H_1 \notin \overline{\mathcal{S}}$ *and* $H_2 \notin \overline{\mathcal{S}}$
   - *(iii)* $H_1 \supseteq_i K_3 + P_1$ *and* $H_2 \supseteq_i 4P_1$ *or* $2P_2$

[2]We refer the reader to Chapter 2 for the notation. Additionally, we add that $\mathcal{S}$ denotes the class of graphs every connected component of which is either a subdivided claw or a path on at least one vertex.

(iv) $H_1 \supseteq_i K_{1,3}$ *and* $H_2 \supseteq_i K_4$ *or* C_4

(v) $H_1 \supseteq_i$ diamond *and* $H_2 \supseteq_i K_{1,3},\ 5P_1,\ P_2 + P_4$ *or* $P_1 + P_6$

(vi) $H_1 \supseteq_i 2P_1 + P_2$ *and* $H_2 \supseteq_i K_3 + P_1,\ K_5,\ \overline{P_2 + P_4}$ *or* $\overline{P_6}$

(vii) $H_1 \supseteq_i K_3$ *and* $H_2 \supseteq_i 2P_1 + 2P_2,\ 2P_1 + P_4,\ 4P_1 + P_2,\ 3P_2$ *or* $2P_3$

(viii) $H_1 \supseteq_i 3P_1$ *and* $H_2 \supseteq_i \overline{2P_1 + 2P_2},\ \overline{2P_1 + P_4},\ \overline{4P_1 + P_2},\ \overline{3P_2}$ *or* $\overline{2P_3}$

(ix) $H_1 \supseteq_i K_4$ *and* $H_2 \supseteq_i P_1 + P_4,\ 3P_1 + P_2$ *or* $2P_2$

(x) $H_1 \supseteq_i 4P_1$ *and* $H_2 \supseteq_i$ gem, $\overline{3P_1 + P_2}$ *or* C_4

(xi) $H_1 \supseteq_i$ gem, $\overline{P_1 + 2P_2}$ *or* $\overline{P_2 + P_3}$ *and* $H_2 \supseteq_i P_1 + 2P_2$ *or* P_6

(xii) $H_1 \supseteq_i P_1 + P_4$ *and* $H_2 \supseteq_i \overline{P_1 + 2P_2}$*, or*

(xiii) $H_1 \supseteq_i 2P_2$ *and* $H_2 \supseteq_i \overline{P_2 + P_4},\ \overline{3P_2}$ *or* $\overline{P_5}$*.*

We prove Theorem 7 by giving two general techniques for constructing atoms of unbounded clique-width (see Lemmas 59 and 60) and by modifying existing graph constructions for proving unbounded clique-width of the whole class [149, 69, 17, 31, 137]. Additionally, our constructions are frequently built on a modification of a k-subdivided wall; which has unbounded clique-width for any $k \geq 0$ [148].

1.4 Graph Deletion Problems

Many standard computational problems, including maximum clique, maximum independent set, or coloring, which are NP-hard in general, have polynomial-time exact or approximation algorithms in restricted graph classes. Such polynomial-time algorithms can sometimes be adjusted to also work on graphs that are "close" to graphs from these classes. Usually, the "closeness" of a graph G to a graph class $\mathcal{G}$ is measured by the number of operations required to transform G into a graph from the class $\mathcal{G}$, where a single operation consists either on removing a vertex from G or on adding or removing an edge from G. Such an approach leads us to the following generic problem.

Graph modification problem into a class of graphs $\mathcal{G}$	
Instance:	A graph G (typically not from $\mathcal{G}$) and a number k
Question:	Can G be transformed into a graph from the class $\mathcal{G}$ by performing $\leq k$ modifications of an appropriate kind?

Depending on the kind of modifications allowed, we obtain four variants of this problem: vertex deletion problem, edge deletion problem, edge completion problem, and edge edition problem (the latter allowing both deletions and additions of edges).

Graph deletion problem into $\mathcal{G}$	
Instance:	A graph G and a number k
Question:	Can G be transformed into a graph from $\mathcal{G}$ by performing $\leq k$ vertex deletions?

Lewis and Yannakakis [144] showed that the vertex deletion problem into any non-trivial hereditary class of graphs is NP-hard. This is not surprising, as many classical hard problems can be formulated as vertex deletion problems into particular classes of graphs, for example, Vertex Cover as vertex deletion to edgeless graphs, Feedback Vertex Set as vertex deletion to forests, and Odd Cycle Transversal as vertex deletion to bipartite graphs.

Graph modification problems are a popular research direction in the study of the *parameterized complexity* of NP-complete problems. We often choose the parameter k as the number of allowed modifications, so the instance of such a problem is the pair (G, k). It turns out that characterizations by forbidden structures are sometimes useful to design FPT algorithms for graph modification problems. For example, Cai [39] proposed an FPT algorithm for modification problems into classes of graphs characterized by a finite family of forbidden induced subgraphs $\mathcal{F}$. His algorithm identifies a forbidden structure in the input graph (which can be done in polynomial time when $\mathcal{F}$ is finite) and branches over all possible ways of modifying that structure.

Let us now consider the class of perfect graphs, graphs where the size of a maximum clique equals the coloring number for every induced subgraph. Grötschel, Lovász, and Schrijver [108] showed that in the class of perfect graphs the maximum clique, the maximum independent set, and the minimum coloring problems can be solved in polynomial time. Perfect graphs include many well-known graph classes that have been particularly intensively studied due to several practical and theoretical applications. Among them are: interval graphs, proper interval graphs, chordal graphs, permutation graphs (intersection graphs of line segments whose endpoints lie on two parallel lines), comparability graphs (graphs whose edges correspond to the pairs of vertices comparable in some fixed partial order $<$ on the vertex set; such an order is called a *transitive orientation* of the graph), and co-comparability graphs. It is well known that the class of permutation graphs corresponds to the intersection of comparability and co-comparability graphs [160] (see Figure 1.2 for the hierarchy of inclusions). All these graph classes are *hereditary* and for all of them, a characterization by forbidden induced subgraphs is known, see [142] for interval graphs, [94] for comparability and permutation graphs.

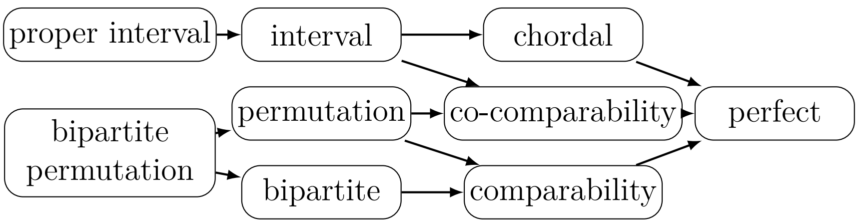


Figure 1.2 [26] Hierarchy of inclusions between graph classes considered in the introduction. An arrow from graph class $\mathcal{A}$ to graph class $\mathcal{B}$ indicates that $\mathcal{A} \subset \mathcal{B}$.

Since the families of forbidden structures are infinite for graph classes introduced above, modification algorithms for these classes have to be much more sophisticated. For several of them, modification problems have satisfactory solutions:

- chordal graphs: all four versions of the modification problem are FPT [44, 154];

- interval graphs: edge completion and edge deletion are FPT [172, 42], vertex deletion is FPT [44], edge edition remains open;

- proper interval graphs: all four versions of the modification problem are FPT [43].

On the other hand, it is known that the vertex deletion to perfect graphs is W[2]-hard [116]. It is worth mentioning that for a long time, it was unknown whether there are classes of graphs recognizable in polynomial time for which modification problems are hard. The first such example was given by Lokshtanov [146], who proved that the vertex deletion is W[2]-hard for graphs avoiding all *wheels* (i.e., cycles with an additional vertex adjacent to all other vertices). It is unknown whether comparability graphs, co-comparability graphs, and permutation graphs have FPT modification algorithms. The class of co-comparability graphs, which constitutes the superclass of interval graphs and an important subclass of perfect graphs, seems to be particularly interesting from the parameterized point of view.

Our focus. We consider the bipartite subclass of permutation graphs, *bipartite permutation graphs*. Like the class of interval graphs, the class of permutation graphs admits polynomial-time algorithms for rich family problems which are NP-complete in general. Apart from the already mentioned classical hard problems which are polynomial-time solvable for perfect graphs, there also exist polynomial algorithms solving e.g., HAMILTONIAN CYCLE, FEEDBACK VERTEX SET or DOMINATING SET in the class of permutation graphs [34, 78]. In light of the above considerations, since all the modification problems into the class of permutation graphs—and the related classes of comparability and co-comparability graphs—remain open, restricting our attention to the class of bipartite permutation graphs appears to be a natural research direction.

Bipartite permutation graphs form an interesting graph class themselves, first investigated by Spinrad, Brandstädt, and Stewart [166], who characterized them by means of appropriately chosen linear orderings of its bipartition classes. One of the most interesting results concerning the bipartite permutation graphs is by Heggernes et al. [117], who showed that the NP-complete problem of computing the *cutwidth* of a graph (i.e., finding a linear order of the vertices of a graph that minimizes the maximum number of edges intersected by any line inserted between two consecutive vertices) is polynomial for bipartite permutation graphs.

Our algorithm exploits the absence of some forbidden structures in bipartite permutation graphs. Since these structures cannot, in particular, occur in permutation graphs, we believe that besides being a complete result itself, our research is a step towards understanding the parameterized complexity of modification problems into permutation graphs.

We focus on the modification by vertex deletion and prove that it is in FPT parameterized by the number of deleted vertices. Our result is summarized in the following theorem.

Theorem 8. *There is an $\mathcal{O}(9^k \cdot |V(G)|^9)$-time algorithm for instances (G, k) of the problem of vertex deletion into bipartite permutation graphs.*

Our algorithm is based on the characterization of bipartite permutation graphs by forbidden subgraphs. Using the characterization, at first, we get rid of constant-size forbidden subgraphs by branching, which is a standard technique in modification problems on hereditary graph classes [123, 172]. We call graphs without these forbidden subgraphs *almost bipartite permutation graphs.* We note that this class may contain holes on more than ten vertices, in contrast to bipartite permutation graphs.

One of our main contributions lies in a structural analysis of almost bipartite permutation graphs. Our approach is partially inspired by the ideas of van 't Hof and Villanger [123] who used similar tools in their work on the proper interval vertex deletion problem. We use the result of Spinrad, Brandstädt, and Stewart [166], who showed that the vertices of every connected bipartite permutation graph $G = (U, W, E)$ can be embedded into a strip in such a way that the vertices from U are on the bottom edge of the strip, the vertices from W are on the top edge of the strip, the neighbors $N(u)$ of u occur consecutively on the top edge of the strip for every $u \in U$ (adjacency property), the vertices from $N(u) - N(u')$ occur consecutively on the top edge of the strip for every $u, u' \in U$ (enclosure property), and the analogous properties are satisfied by the vertices in W (see Figure 1.3).

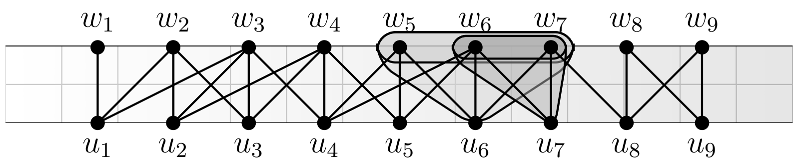


Figure 1.3 [26] Embedding of a bipartite permutation graph (U, W, E) into a strip satysfying the adjacency and the enclosure properties.

Our structural result asserts that, depending on the parity of the length of the shortest hole, a connected almost bipartite permutation graph may be naturally embedded either in a cylinder, or a Möbius strip, locally satisfying adjacency and enclosure properties (see Figure 1.4).

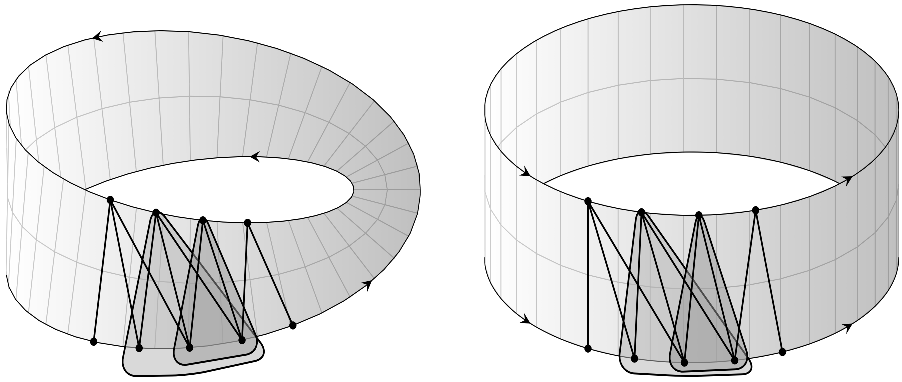


Figure 1.4 [26] An embedding of a connected almost bipartite permutation graph in a cylinder or a Möbius strip that locally satisfies the adjacency and enclosure properties.

Once we obtain such a structure, we show that every minimal vertex cut that destroys all holes lies near a few consecutive vertices from the shortest hole. This allows us to check all the possibilities where we can find a minimum cut. Finally, we use a polynomial algorithm for finding maximum flow (and thus a minimum cut).

This approach to proving Theorem 8 can be slightly modified to obtain a 9-approximation algorithm for the bipartite permutation vertex deletion problem; see [26] for the details.

A natural follow-up question to this work is whether the problem of Vertex Deletion into Bipartite Permutation Graphs admits a polynomial kernel. It has been resolved in the affirmative by Derbisz, Kanesh, Madathil, Sahu, Saurabh, and Verma in [79]. They designed a polynomial kernel of size $\mathcal{O}(n^{229})$ and claimed that by more precise case analysis one should get a size about $\mathcal{O}(n^{100})$.

1.5 Tuza's Conjecture for Threshold Graphs

We are now moving to the area of packing and covering.

> If we can "pack" at most k disjoint objects of some type in a given graph, how many elements do we need to "cover" all appearances of such an object in the graph?

Erdős and Pósa famously proved that if a graph contains at most k pairwise vertex-disjoint cycles, then there is a set of at most $f(k)$ vertices that intersects every cycle [85]. While the exact best value of function f is yet unknown, the asymptotic behavior was recently determined to be $f(k) = \Theta(k \log k)$ [41].

In this work, we focus on edge-disjoint triangles; we refer the interested reader to [163] for a dynamic survey on other objects. For a graph G, we call every family of pairwise edge-disjoint triangles a *triangle packing*, and every subset of edges intersecting all triangles in G a *triangle hitting*. We denote by $\mu(G)$ the maximum size of a triangle packing in G, and by $\tau(G)$ the minimum size of a triangle hitting in G. Trivially, there is a set of at most $3\mu(G)$ edges that intersect every triangle. We are concerned with improving that bound, following Tuza's conjecture from 1981.

Conjecture 1 (Tuza [170])**.** *For any graph G it holds $\tau(G) \leq 2\mu(G)$.*

Conjecture 1, if true, is tight for K_4 and K_5. Gluing together copies of K_4 and K_5 along vertices, it is easy to build an infinite family of connected graphs for which Conjecture 1 is tight. However, for larger cliques, it is known that the ratio $\tau(K_p)/\mu(K_p)$ tends to $3/2$ as p increases [87]. In addition, Haxell and Rödl [115] proved that $\tau(G) \leq 2\mu(G) + o(|V(G)|^2)$ for any graph G, meaning Conjecture 1 is asymptotically true when $\tau(G)$ is quadratic with respect to $|V(G)|$. Those seem to indicate that Conjecture 1 should be easier for dense graphs than for sparse graphs. Conversely, it is asymptotically tight in some classes of dense graphs [13]. If we focus on *hereditary graph* classes (i.e. classes that contain every induced subgraph of a graph in the class), the conjecture has only been confirmed for a few graph classes. Those classes include most notably graphs of treewidth at most 6 [22], 4-colourable graphs [141], and graphs with maximum average degree less than 7 [161].

A good candidate for an interesting hereditary graph class is the class of *split graphs*. However, Conjecture 1 remains a real challenge even when restricted to split graphs. Another good candidate is the class of *cographs*. As an initial step, we focus on graphs that are both split graphs and cographs, i.e. *threshold* graphs. While this may seem like a small step, it is the first non-trivial hereditary superclass of cliques where the conjecture is confirmed.

Theorem 9. *If G is a threshold graph, then $\tau(G) \leq 2\mu(G)$.*

We work with the representation of threshold graphs where vertices can be split into a clique and independent set with a nested neighborhood in between, i.e., the neighborhood of vertices of any part forms a chain with respect to set inclusion. We split the proof of Theorem 9 into several cases and design particular triangle packings for them separately. One of the main ideas lies in case splitting: We look at the first half of the clique (in the order given by nested adjacencies) and mark the first vertex u_r of the independent set that sees all the vertices there (see Figure 1.5 for the illustration). By that, we make sure that all the following vertices are complete to the first half of the clique. We then split the analysis based on the number of such vertices compared to half of the independent set size.

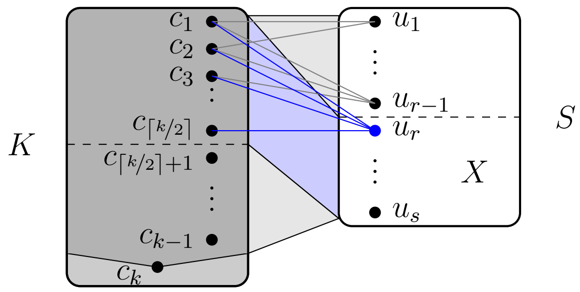


Figure 1.5 The structure of threshold graph G with $G[K]$ being a clique, $G[S]$ independent set. $N(c_{i+1}) \subseteq N(c_i)$ for all $1 \leq i < k$ and $N(u_i) \subseteq N(u_{i+1})$ for all $1 \leq i < s$. $r \in \{1, \dots, s\}$ is chosen to be minimal such that $\{c_1, \dots, c_{\lceil k/2 \rceil}\} \subseteq N(u_r)$.

Additionally, we show that similar tools with more involved analysis can be used to verify Conjecture 1 also for specific co-chain graphs, that is when both sides are of the same size that is divisible by four.

Theorem 10. *If G is an even balanced co-chain graph, then $\tau(G) \leq 2\mu(G)$.*

Theorem 10 can be seen as a very first step towards attacking Conjecture 1 on (mixed) unit interval graphs as those graphs can be modeled as a *concatenation* of co-chain graphs. That is, vertices of graph G are partitioned into r cliques $C_1, \dots, C_r$ where each (C_i, C_{i+1}) induce a co-chain graph and G contains no other edges; see [118, 136] for more details. The simplest object for further study might be a k-path, which can be viewed as a concatenation of well-structured same-sized co-chain graphs.

Finally, it is worth mentioning that Conjecture 1 is known to hold as soon as we consider *multi*-packing [45], and in particular it holds in its fractional relaxation [138]. Another angle of attack consists of lowering the bound of 3 step by step for all graphs. The best, and in fact only, such bound is slightly under 2.87 [114].

1.6 Organization of the Thesis

Chapter 3 contains the study of $S_{t,t,t}$-free graphs. The formal definition of extended strip decomposition is given in Section 3.1. The proof of the main structural result, Theorem 2, is given in Section 3.2, and its applications, Theorems 4 and 3, in Section 3.3. Section 3.4 mentions other applications and the improvement of Theorem 2.

Chapter 4 contains the structural results of $(2P_4, C_5)$-free graphs develop to design a polynomial algorithm for 3-Coloring in this class of graphs. We give additional motivation as to why we forbid an induced cycle in Section 4.1. The proof of Theorem 5 is written in Section 4.2; consisting of a proof summary in the beginning, proof of the cases where our graph is additionally C_7-free in Section 4.2.1, and the main part of the proof (case (ii)) consisting of more complicated reductions in Section 4.2.2 followed up by the proof in Section 4.2.3.

Chapter 5 contains the study of boundedness of clique-width of (H_1, H_2)-free atoms. We prove Theorem 6 in Section 5.3. We prove Theorem 7 in Section 5.6, with unbounded constructions in Section 5.5.

Chapter 6 contains the study of vertex deletion to bipartite permutation graphs. Section 6.1 contains additional notation, concerning mainly notations from the world of posets. We prove the main structural result in Section 6.2 which is then used in the proof of the main theorem, Theorem 8 in Section 6.3. We conclude with a few open questions in Section 6.4.

Chapter 7 confirms the Conjecture 1 for threshold graphs, in Section 7.2, and for co-chain graphs with both sides of the same size divisible by four, in Section 7.3. Before the actual proofs, Section 7.1 contains notes on triangle packings of complete graphs and complete split graphs that are used heavily in the proofs.

1.7 Covered papers

Results included in this thesis come from the five papers listed below which are joint publications with other colleagues. I want to emphasize that I actively contributed to all the standard stages of the creation process in each paper; particularly to developing the ideas as well as preparing the write-up including writing the results, revising and refining preliminary drafts, and preparing the final versions for publication.

- *Max Weight Independent Set in graphs with no long claws: An analog of the Gyárfás' path argument* [150], published in ACM Transactions on Computation Theory, with a conference version presented at ICALP 2022 [151][3]. The paper covers the structural result for $S_{t,t,t}$-free graphs (Theorem 2) and its applications, the subexponential-time algorithm (Theorem 3) and QPTAS (Theorem 4) for the Maximum Weight Independent Set problem, presented in Chapter 3. Small corrections of [150] are implemented. The main structural result was developed in a joint discussion of all the authors. In the final write-up, I collaborated in Section 3.2.

- *On 3-Coloring of* $(2P_4, C_5)$*-Free Graphs* [129], published in Algorithmica, with a conference version presented at WG 2021 [128]. This paper covers the polynomial algorithm for 3-Coloring in $(2P_4, C_5)$-free graphs, presented in Chapter 4. I introduced the topic to my coauthors as well as the initial ideas on how to tackle the problem. The algorithm was then developed in a joint discussion of all the authors.

[3] I would like to note that during my doctoral studies I changed my surname from Novotná to Masaříková, and as a result, some of my publications appear under my former name.

- *Clique-width: Harnessing the power of atoms* [75], published in the Journal of Graph Theory, with a conference version presented at WG 2020 [74]. The paper comprises numerous smaller results—unbounded constructions for clique-width of (H_1, H_2)-free atoms— as well as the main structural result detailed in Theorem 6. The results are presented in Chapter 5. I contributed approximately one-third of the ideas in the proof of the main positive theorem, Theorem 6, and provided insights for several key constructions.

- *Vertex Deletion into Bipartite Permutation Graphs* [26], published in Algorithmica, with a conference version presented at IPEC 2021 [25] (obtaining best paper and best student paper awards). The result originated from a workshop problem that I worked on with Karolina Okrasa and Łukasz Bożyk. Together, we solved Theorem 8, with each of us contributing equally to the ideas. My ideas led to an understanding of the structure of connected almost bipartite permutation graphs, namely, I proved Propositions 80 and 81 that served as a starting point for us to develop the entire structure together.

  While writing up our results, we were contacted by Jan Derbisz and Tomasz Krawczyk, who independently solved the problem, initially framed in terms of partially ordered sets. Seeing the similarity of our methods, we decided to join forces for the write-up with me coordinating the group then.

- *Tuza's Conjecture for Threshold Graphs* [19], published in Discrete Mathematics & Theoretical Computer Science, with a conference version presented at EUROCOMB 2021 [18]. The paper addresses Chapter 7 focusing on the verification of Conjecture 1 for threshold graphs (Theorem 9) and a subclass of co-bipartite chain graphs (Theorem 10). My contribution is the largest among the authors. I contributed significantly to ideas leading to solutions and also to their simplification, in particular, I developed the way how both main proofs are split into the cases in the current paper. Moreover, I led and coordinated the project, and wrote about a third of the paper, including Section 7.3 which contains one of the two main proofs, proof of Theorem 10.

1.8 Other results

During my doctoral studies, I obtained other results that I did not include in my thesis. Let me describe them briefly here.

Bounding the number of minimal separators. In paper [93], we confirm a conjecture of Gartland and Lokshtanov (originally in [96], modified in [98]) which states that for every natural number k, the family $\mathcal{C}$ of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor is *tame*, that is, every graph in $\mathcal{C}$ on n vertices contains at most $n^{\mathcal{O}(1)}$ minimal separators. By a result of Fomin, Todinca, and Villanger [91] the latter entails the existence of polynomial-time algorithms for MWIS, Feedback Vertex Set, and many other problems, when restricted to an input graph from G.

Furthermore, as shown by Gartland and Lokshtanov, our result implies a full dichotomy of hereditary graph classes defined by a finite set of forbidden induced subgraphs into tame (admitting a polynomial bound of the number of minimal separators) and feral (containing infinitely many graphs with exponential number of minimal separators). Specificaly, it follows that for a graph family $\mathcal{C}$ defined by a finite number of forbidden induced subgraphs it holds that if there exists a natural number k such that $\mathcal{C}$ forbids all k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs [4], then $\mathcal{C}$ is tame. Otherwise $\mathcal{C}$ is feral. Our proof heavily build on [96] where the authors provided a weaker statement of having a quasipolynomial bound on the number of minimal separators, rather than a polynomial bound.

Balanced separator for planar hyperbolic graphs. In paper [133], we studied planar graphs of bounded hyperbolicity. Let us define an alternative notion of slimmness, as it gives a quicker intuition for graphs than hyperbolicity and it is known that these two notion differ by a constant factor from each other. Consider three vertices x, y, z and take any three shortest paths connecting all three pairs of the vertices; calling the shortest paths *sides* of the *triangle* xyz. We say that the graph is δ-slim if for each triangle the side xy is within the distance δ from the union of the sides yz and zx. The limness of a graph is then the smallest such δ. Let us point out here that the class of planar δ-hyperbolic graphs is not hereditary, as illustrated, for instance, by removing the middle vertex from a wheel (thus, obtaining a cycle).

Our main contribution lies in providing a balanced separator for planar δ-hyperbolic graphs that is additionally well-structured: the separator is either a geodesic cycle or a geodesic path. Specifically, it follows from our proofs that for any $\delta \geq 0$, the class of connected planar δ-hyperbolic graphs has a $1/2$-balanced in-class separator of size $2^{\mathcal{O}(\delta)} \log n$ that can be computed in $2^{\mathcal{O}(\delta)} n \log^5 n$ time.

Compared to using the well-known separator theorem by Lipton and Tarjan for planar graphs or the separator derived from bounded treewidth of δ-hyperbolic graphs, the crucial advantage is that our separator is *in-class*, meaning that any connected component along with the separator forms again a planar δ-hyperbolic graph. This property makes it suitable for use in recursive algorithms. We subsequently presented applications of our separator theorem, namely a near-linear time FPTASes for Maximum Independent Set and Traveling Salesman (which aims to find a shortest closed walk in the graph that visits every vertex at least once). Additionally, we demonstrated that our approximation scheme for Maximum Independent Set achieves the best possible running time under the ETH.

List locally surjective homomorphism problem in F-free graphs. In paper [81] we studied the complexity of the list locally surjective homomorphism problem in F-free graphs for a fixed graph F. A locally surjective homomorphism from a graph G to a graph H (with possible loops) is an edge-preserving mapping $h : V(G) \rightsquigarrow V(H)$ that is surjective in the neighborhood of each vertex in G, that is, if $h(v) = a \in V(H)$, then for every neighbor b of a in H (including a, if H has a loop) there is a neighbor u of v in G, such that

[4] We refer the reader to Fig. 2 [98] for the definitions of k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs)

$h(u) = b$. In the list locally surjective homomorphism problem (LLSHOM(H)), the graph H is fixed and the instance consists of a graph G whose every vertex is equipped with a subset of $V(H)$, called *list*. We ask for the existence of a locally surjective homomorphism from G to H, where every vertex of G is mapped to a vertex from its list. We studied for which pairs of (H, F) the problem LLSHOM(H) in F-free graphs can be solved in subexponential time concentrating mainly on F being a forest whose every component is a path or a subdivided claw. We obtained a positive result for H being P_3 or C_4. Complementing the dichotomy by proving that there is t, such that the LLSHOM(H) cannot be solved in subexponential time in P_t-free graphs, unless the ETH fails. Moreover, we added a reduction showing for all graphs H, for which LLSHOM(H) is NP-hard in general graphs (that is H is not a vertex, a vertex with a loop, an edge), LLSHOM(H) cannot be solved in subexponential time in F-free graphs for F being a bounded-degree forest, unless the ETH fails.

Robust Connectivity of Graphs on Surfaces. We study the notion of *robust connectivity* of a graph, introduced recently in [28] as a tool to show flexible choosability results. The *robust connectivity* of a graph, $\kappa_\rho(G)$, is defined as the minimum value $\frac{|R \cap \Lambda(T)|}{|R|}$ taken over all nonempty subsets $R \subseteq V(G)$, where $\Lambda(T')$ denote the set of leaves in a tree T' and $T = T(R)$ is a spanning tree on G chosen to maximize $|R \cap \Lambda(T)|$. We note that the robust connectivity is related to a well-known NP-hard problem MAXIMUM LEAF NUMBER, which asks to maximize $\Lambda(T)$ over all spanning trees T. In our paper [27], we investigate robust connectivity for graphs embedded in surfaces. We prove a tight asymptotic bound of $\Omega(\gamma^{-\frac{1}{r}})$ for the robust connectivity of r-connected graphs of Euler genus γ. Moreover, we give a surprising connection between the robust connectivity of graphs with an edge-maximal embedding in a surface and the *surface connectivity* of that surface, which describes to what extent large induced subgraphs of embedded graphs can be cut out from the surface without splitting the surface into multiple parts. For planar graphs, this connection provides an equivalent formulation of a long-standing conjecture of Albertson and Berman [6], which states that every planar graph on n vertices contains an induced forest of size at least $n/2$.

2 Preliminaries

We first give a general graph terminology and notation.

2.1 Basic Graph Terminology and Notation

Let $G = G(V, E)$ be a simple graph, i.e., undirected, with no loops and parallel edges. Unless stated otherwise, all graphs considered in this work are simple, and we measure their size by the number of vertices, considering $n = |V(G)|$.

We say that *vertices* $u, v \in V(G)$ are *adjacent* whenever $uv \in E(G)$ (that is, uv is an edge of G), otherwise, they are *non-adjacent*. For a subset $S \subseteq V(G)$, the subgraph of G *induced by* S is the graph $G[S]$, which has vertex set S and edge set $\{uv \mid uv \in E(G), u, v \in S\}$. We say that S in an *induced subgraph* of G. If $S = \{s_1, \ldots, s_r\}$, we may write $G[s_1, \ldots, s_r]$ instead of $G[\{s_1, \ldots, s_r\}]$. If the graph G is clear from the context, we will often identify induced subgraphs with their vertex sets. We write $F \subseteq_i G$ to indicate that F is an induced subgraph of G. For a subset $S \subseteq V(G)$, we let $G \setminus S = G[V(G) \setminus S]$.

A graph H is a *minor* of a graph G, if H can be obtained from a subgraph of G by a sequence of edge contractions; where the contraction of edge uv in $G(V, E)$ provides the graph $G'(V', E')$ with $V' = V \setminus \{v\}$ and $E' = E \setminus \{e \mid v \in e\} \cup \{uw \mid; vw \in E, u \neq w\}$. If H was derived from an induced subgraph of G, we call H an *induced minor*.

The *(open) neighborhood* of a vertex $u \in V(G)$ is the set $N_G(u) = \{v \in V(G) \mid uv \in E(G)\}$. Two vertices in G are *false twins* if they have the same neighborhood; note that such vertices must be non-adjacent. The *closed neighbourhood* of a vertex $u \in V(G)$ is the set $N_G[u] = N(u) \cup \{u\}$. For a set $X \subseteq V(G)$, we also define $N_G(X) \coloneqq \bigcup_{v \in X} N_G(v) \setminus X$, and $N_G[X] = N_G(X) \cup X$. If it does not lead to confusion, we omit the subscript and write simply $N(\cdot)$ and $N[\cdot]$.

A *(connected) component* of G is a maximal connected subgraph of G. By $\mathsf{cc}(G)$ we denote the set of connected components of G. A set $S \subset V(G)$ is a u, v-separator if u and v are in distinct components of $G \setminus S$. The set S is a u, v-minimal separator if S is a u, v-separator, but no proper subset of S is a u, v-separator. Finally, S is a *separator* (alternatively a *cut-set*) , respectively *minimal* separator, if S is a u, v-separator, respectively u, v-minimal separator, for some pair of vertices u and v. We say that the separator S is balanced if there exists a constant α, such that the number of vertices in any connected component of $G \setminus S$ is at most αn. When a separator has a specific structure, we may refer to it accordingly; for example, a *clique* separator is one that induces a clique in G.

Let X and Y be two disjoint vertex subsets of a graph G. The edges between X and Y form a *matching* if every vertex in X is adjacent to at most one vertex in Y and vice versa. A vertex $x \in V(G) \setminus Y$ is *(anti-)complete* to Y if it is (non-)adjacent to every vertex in Y. Similarly, X is *complete* to Y if every vertex of X is complete to Y and *anti-complete* to Y if every vertex of X is anti-complete to Y. Note that this, in particular, implies that X and Y are disjoint. We say that two sets X, Y *touch* if $X \cap Y \neq \emptyset$ or there is an edge with one end in X and another in Y. A vertex $u \in V(G)$ is *dominating* if it is complete to $V(G) \setminus \{u\}$. A set $A \subseteq V(G)$ dominates $B \subseteq V(G)$, $A \cap B = \emptyset$ if every vertex in B has a neighbor in A.

For $k \geq 1$, a *k-subdivision* of G is the operation of replacing each edge uv of G with a $(k+1)$-edge path, whose end-vertices are identified with u and v, respectively.

The *complement* $\overline{G}$ of G has vertex set $V(\overline{G}) = V(G)$ and edge set $E(\overline{G}) = \{uv \mid u, v \in V(G), u \neq v, uv \notin E(G)\}$. The *line graph* of G is the graph with vertex set $E(G)$ and an edge between two vertices e_1 and e_2 if and only if e_1 and e_2 share a common end-vertex in G. For two graphs H_1 and H_2, we let $H_1 + H_2$ denote their disjoint union, and we write kH for the disjoint union of k copies of a graph H.

An *independent set* in a graph G is a subset of $V(G)$ that consists of pairwise non-adjacent vertices; a *clique* in G is a subset of pairwise adjacent vertices.

2.2 Forbiden Induced Subgraph Notations

Let H be a graph. A graph G is *H-free* if G does not contain H as an induced subgraph. Let $\mathcal{H} = \{H_1, \ldots, H_p\}$ be a set of graphs. Then G is *$(H_1, \ldots, H_p)$-free* (or $\mathcal{H}$-free) if it is H_i-free for all $i \in \{1, \ldots, p\}$. If $|\mathcal{H}| = 1$ or $|\mathcal{H}| = 2$, then the class of $\mathcal{H}$-free graphs is said to be *monogenic* or *bigenic*, respectively.

2.3 Specifically Named Graphs and Graph Classes

Here we define graph classes and specific named graphs mentioned in the Introduction and throughout the thesis, unless it is very specific to only one chapter of the thesis.

The graphs C_t, K_t, and P_t denote the cycle, complete graph, and path on t vertices, respectively. We refer to P_t also as the path of length t, in particular note that we count the number of vertices (and not edges). By $C_{>t}$ we mean all induced cycles of length more than t. The graph $K_{s,t}$ denotes the complete bipartite graph whose two partition classes contain s and t vertices, respectively.

We mention here specific named graphs:

- *hole* is an induced cycle on at least five vertices. We say that a hole is *even* (or *odd*) if it contains an even (odd) number of vertices, respectively;

- *claw* is the graph $K_{1,3}$, i.e., graph consisting of vertices x, y_1, y_2, y_3 and edges xy_i for $i \in \{1, 2, 3\}$;

- *subdivided claw* $S_{h,i,j}$, for $1 \leq h \leq i \leq j$ is the tree with one vertex x of degree 3 and exactly three leaves, which are of distance h, i and j from x, respectively;

- *star* is the graph $K_{1,s}$ for $s \geq 3$, i.e., one vertex (*center vertex*) adjacent to all the others.

- *paw* is the graph $\overline{P_1 + P_3}$;

- *diamond* is the graph $\overline{2P_1 + P_2}$;

- *gem* is the graph $\overline{P_1 + P_4}$.

- *k-skinny-ladder* is a graph G consisting of two anti-complete paths $P_\ell = \ell_1, \ell_2, \ldots, \ell_k$ and $P_r = r_1, r_2, \ldots r_k$ and a set $\{s_1, s_2, \ldots, s_k\}$ of vertices such that for every i, s_i is adjacent to ℓ_i and r_i and to no other vertices.

- *k-creature* is a graph consisting of four disjoint vertex sets A, B, $X = \{x_1, \ldots x_k\}$, $Y = \{y_1, \ldots y_k\}$ such that: (a) A is connected and B is connected, (b) there are no edges from A to $Y \cup B$ and no edges from B to $X \cup A$, (c) A dominates X and B dominates Y and (d) $x_i y_j$ is an edge if and only if $i = j$.

A graph is:
- *bipartite* if its vertex set can be partitioned into two (possibly empty) independent sets.
- *complete multi-partite* if its vertex set can be partitioned into r independent sets $V_1, \ldots, V_r$ for some integer $r \geq 1$ such that V_i is complete to V_j for every pair i, j with $1 \leq i < j \leq r$; if $r = 2$, we say that the graph is *complete bipartite*.
- *cograph* if it is P_4-free. Alternatively, it can be constructed from a single vertex by complementation and disjoint union operations.
- *chordal* graph if it has no induced cycles on more than four vertices, that is, if it is $(C_4, C_5, \ldots)$-free. Equivalently, chordal graphs are intersection graphs of subtrees of a tree.
- *co-chordal* graph if its complement is chordal.
- *split* graph if its vertex set can be partitioned into a clique and an independent set.
- (*bipartite*) *chain* graph if it is bipartite, say with bipartition classes X and Y, such that the vertices of X can be ordered $x_1, \ldots, x_p$ with the property that $N(x_1) \subseteq N(x_2) \subseteq \ldots \subseteq N(x_p)$.
- *co-chain* graph (or sometimes alternatively called *co-difference graph*) if it is the complement of a bipartite chain graph.
- *interval* graph if it is an intersection graph of intervals on a real line.
- *proper interval* graph if it is an intersection graph of intervals none of which is contained in another.
- *permutation* graph if it is an intersection graph of line segments whose endpoints lie on two parallel lines.
- *comparability* graph if its edges correspond to the pairs of vertices comparable in some fixed partial order $<$ on the vertex set (such an order is called a *transitive orientation* of the graph).
- *co-comparability* graph if it is the complement of a comparability graph.

2.4 Graph parameters

The *clique-width* of a graph G, denoted by $\text{cw}(G)$, is the minimum number of labels needed to construct G using the following four operations:

1. create a new graph consisting of a single vertex v with label i;

2. take the disjoint union of two labeled graphs G_1 and G_2;

3. add an edge between every vertex with label i and every vertex with label j ($i \neq j$);

4. relabel every vertex with label i to have label j.

A class of graphs $\mathcal{G}$ has *bounded* clique-width if there is a constant c such that $\mathrm{cw}(G) \leq c$ for every $G \in \mathcal{G}$; otherwise the clique-width of $\mathcal{G}$ is *unbounded.*

2.5 Computational Complexity

In this thesis, we distinguish a few qualitatively different running time categories which provide a finer classification of problems that are more tractable under some circumstances. First, we focus on a parameterized approach. Given a parameterized problem with parameter k and the input size n, we say that it is *fixed-parameter tractable (*FPT*)* if there is an algorithm running in time $f(k)n^c$ for some computable function f and constant c. This is deemed an efficient algorithm in a parameterized realm. Slightly less efficient is a *slice-wise polynomial (*XP*)* problem for which there is an algorithm runnning in time $n^{f(k)}$. However, notice that such an algorithm is still polynomial when the parameter k is constant.

If we do not have access to a parameter, additionally to algorithms running in a polynomial time, we distinguish *quasipolynomial-time* algorithms which are running in time $2^{\mathcal{O}(\log^c(n))}$ for some constant c and the input size n. Note that this class is polynomial when $c = 1$. The last resort is to at least look for a *subexponential* algorithm. Those run in time $2^{o(n)}$. Note that it is better than the conjectured running time to solve the 3-SAT problem which serves as a base for conditional lower-bounds assuming the ETH.

Conjecture 2 (Exponential Time Hypothesis (ETH) [67])**.** *There exists $c > 1$ such that no algorithm solves 3-SAT with n variables in time c^n.*

Whenever in the thesis, if the base of a logarithmic function is not specified, we mean the logarithm of base 2, i.e., $\log n \coloneqq \log_2 n$.

3 Max Weight Independent Set in graphs with no long claws: An analog of the Gyárfás' path argument

We revisit recent developments for the Maximum Weight Independent Set problem in graphs excluding a subdivided claw $S_{t,t,t}$ as an induced subgraph [53] and provide a subexponential-time algorithm with improved running time $2^{\mathcal{O}(\sqrt{nt}\log n)}$ and a quasipolynomial-time approximation scheme with improved running time $2^{\mathcal{O}(\varepsilon^{-1}t\log^5 n)}$. Recall the Gyárfás' path argument for P_t-free graphs: in polynomial time we can find a set P of at most $t-1$ vertices, such that every connected component of $G-N[P]$ has at most $n/2$ vertices. Our main technical contribution is an analog of the Gyárfás' path argument for $S_{t,t,t}$-free graphs: given an n-vertex $S_{t,t,t}$-free graph, in polynomial time we can find a set P of $\mathcal{O}(t\log n)$ vertices and an extended strip decomposition of $G-N[P]$ such that every particle of the said extended strip decomposition has at most $n/2$ vertices. Thus, the extended strip decomposition plays a role of an appropriate analog of the decomposition into connected components and a particle presents an appropriate analog of a connected component to recurse on.

After additional preliminaries in Section 3.1, we prove Theorem 2 in Section 3.2. Proofs of Theorems 3 and 4 are provided in Section 3.3. Finally, we discuss future steps in Section 3.4.

3.1 Preliminaries

Notation. Here, we define additional notation to Chapter 2 specific to this chapter. For a family $\mathcal{Q}$ of sets, by $\bigcup\mathcal{Q}$ we denote $\bigcup_{Q\in\mathcal{Q}} Q$. For a function $\mathfrak{w}: V\to\mathbb{Z}_{\geq 0}$ and subset $V'\subseteq V$, we denote $\mathfrak{w}(V') \coloneqq \sum_{v\in V'}\mathfrak{w}(v)$.

By $T(G)$, we denote the set of all triangles in G. Similarly to writing $xy\in E(G)$, we will write $xyz\in T(G)$ to indicate that $G[\{x,y,z\}]\simeq K_3$.

Extended strip decompositions. Now let us define a certain graph decomposition which will play an important role. An *extended strip decomposition* of a graph G is a pair (H,η) that consists of:

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

which satisfy the following properties (also see Figure 3.1):

1. $\{\eta(o)\mid o\in V(H)\cup E(H)\cup T(H)\}$ is a partition of $V(G)$,

2. for every $x \in V(H)$ and every distinct $y, z \in N_H(x)$, the set $\eta(xy, x)$ is complete to $\eta(xz, x)$,

3. every $uv \in E(G)$ is contained in one of the sets $\eta(o)$ for $o \in V(H) \cup E(H) \cup T(H)$, or is as follows:
   - $u \in \eta(xy, x), v \in \eta(xz, x)$ for some $x \in V(H)$ and $y, z \in N_H(x)$, or
   - $u \in \eta(xy, x), v \in \eta(x)$ for some $xy \in E(H)$, or
   - $u \in \eta(xyz)$ and $v \in \eta(xy, x) \cap \eta(xy, y)$ for some $xyz \in T(H)$.

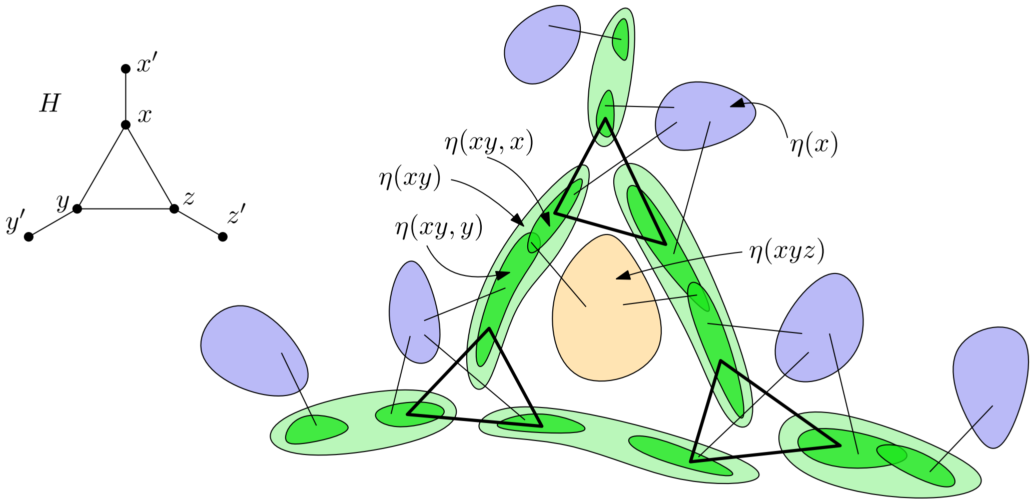


Figure 3.1 A graph H and an extended strip decomposition (H, η) of a graph G. Sets $\eta(\cdot)$ corresponding to vertices, edges, and the triangle of H are marked green, blue, and orange, respectively. The edges between distinct sets are drawn thick if they must exist, and thin if they may exist.

Note that for an extended strip decomposition (H, η) of a graph G, the number of vertices of H can be much larger than the number of vertices of G. However, in such case many sets $\eta(\cdot)$ are empty and thus H is "unnecessarily complicated." An extended strip decomposition (H, η) is *rigid* if (i) for every $xy \in E(H)$ it holds that $\eta(xy, x) \neq \emptyset$, and (ii) for every $x \in V(H)$ such that x is an isolated vertex it holds that $\eta(x) \neq \emptyset$. Observe that if we *restrict* η to $V' \subset V(G)$, i.e. we keep in η only vertices of V', (H, η) after the restriction remains an extended strip decomposition, but it might not be rigid anymore.

We say that a vertex $v \in V(G)$ is *peripheral* in (H, η) if there is a degree-one vertex x of H, such that $\eta(xy, x) = \{v\}$, where y is the (unique) neighbor of x in H. For a set $Z \subseteq V(G)$, we say that (H, η) is an *extended strip decomposition of* (G, Z) if H has $|Z|$ degree-one vertices and each vertex of Z is peripheral in (H, η).

The following theorem by Chudnovsky and Seymour [56] is a slight strengthening of their celebrated solution of the famous *three-in-a-tree* problem. We will use it as a black-box to build extended strip decompositions.

Theorem 11 (Chudnovsky, Seymour [56, Section 6]). *Let G be an n-vertex graph and consider $Z \subseteq V(G)$ with $|Z| \geq 2$. There is an algorithm that runs in time $\mathcal{O}(n^5)$ and returns one of the following:*

- *an induced subtree of G containing at least three elements of Z,*
- *a rigid extended strip decomposition (H, η) of (G, Z).*

Let us point out that actually, an extended strip decomposition produced by Theorem 11 satisfies more structural properties, but of our purpose, we will only use the fact that it is rigid.

Particles of extended strip decompositions. Let (H, η) be an extended strip decomposition of a graph G. We introduce some special subsets of $V(G)$ called *particles*, divided into five *types*.

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

Observe that the number of all particles of (H, η) is at most $\mathcal{O}(|V(H)|^3)$. However, the number of nonempty particles if (H, η) is rigid, is linear in the number of vertices of G.

Observation 12. Let (H, η) be a rigid extended strip decomposition of an n-vertex graph. Then the number of nonempty particles of (H, η) is bounded by $4n$.

Proof. In any extended strip decomposition (H, η), there can be at most $n' \leq n$ nonempty edges of H where each participates in at most four nonempty particles (namely edge interior, full edge, and both half-edge particles). By rigidity of (H, η) we know that there are no empty edges in H. Then there is at most $n - n'$ nonempty vertices and triangles of H where each participates in one nonempty vertex or triangle particle. $\square$

A vertex particle A_x is *trivial* if x is an isolated vertex in H. Similarly, an extended strip decomposition (H, η) is *trivial* if H is an edgeless graph. The following observation follows immediately from the definitions of an extended strip decomposition and particles.

Observation 13. Let (H, η) be an extended strip decomposition of a graph G. For each $xy \in E(H)$ the following hold:

1. $A_{xy}^{\perp} \subseteq A_{xy}^{x} \subseteq A_{xy}^{xy}$,
2. for any $v_x \in \eta(xy, x)$ and $v_y \in \eta(xy, y)$ we have $N(A_{xy}^{xy}) = N(v_x) \cup N(v_y) \setminus A_{xy}^{xy}$.

We conclude this section by recalling an important property of particles of extended strip decompositions, observed by Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [51, 53].

Theorem 14 (Chudnovsky et al. [51, Lemma 6.8])**.** *Let (H, η) be an extended strip decomposition of G. Suppose P_1, P_2, P_3 are three induced paths in G that do not touch each other, and moreover each of P_1, P_2, P_3 has an endvertex that is peripheral in (H, η). Then in (H, η) there is no particle that touches each of P_1, P_2, P_3.*

3.2 Main Result

In this section, we prove our main result:

Theorem 2. *Given an n-vertex graph G with nonnegative vertex weights and integer $t \geq 1$, one can in polynomial time either:*

- *output an induced copy of $S_{t,t,t}$ in G, or*
- *output a set $\mathcal{P}$ consisting of at most $11 \log n + 8$ induced paths in G, each of length at most $t+1$ and a rigid* extended strip decomposition *of $G - N[\bigcup_{P \in \mathcal{P}} V(P)]$ whose every particle has weight at most half of the total weight of $V(G)$.*

Let us first give an overview of our approach. We present a recursive algorithm that, for a given graph G, will return one of the outcomes of Theorem 2. Let w be the total weight of G; the value of w will not change throughout the recursive steps of the algorithm. Note that the *weight* of a (sub)graph is defined as a sum of the weights of its vertices. We start with finding a Gyárfás path Q navigating towards the component of the largest weight in G. That is, by Theorem 1, we find Q such that each connected component of $G - N[Q]$ is of weight at most $\frac{w}{2}$. Finding such small connected components is a great outcome as we can readily include each as a small trivial vertex particle of an extended strip decomposition we are constructing. We say that a particle is *small* if its weight is at most $\frac{w}{2}$, and an extended strip decomposition is *refined* if all its particles are small. Observe that if $|Q| \leq 3t+1$, we immediately get the desired refined trivial extended strip decomposition of $G - N[Q]$. Otherwise, we proceed to the main part of the algorithm. At each step, we will remove some vertices from Q, and will measure the progress of our algorithm in the number of the remaining vertices of Q.

Formally, we create a set $\mathcal{Q}$ of at most two non-touching induced paths such that $\bigcup \mathcal{Q} \subseteq Q$. At each step of recursion, we obtain a set $\hat{\mathcal{Q}}$ of at most two non-touching induced paths with $|\bigcup \hat{\mathcal{Q}}| \leq \frac{2}{3}|\bigcup \mathcal{Q}|$ that takes a role of $\mathcal{Q}$ in the next step of recursion. Hence, in $11 \log n$ recursive steps, $|\bigcup \mathcal{Q}|$ drops below $3t+1$. In the base case of the recursion, when $|\bigcup \mathcal{Q}| \leq 3t+1$, we return the refined trivial extended strip decomposition ensured by maintaining the property that $G - N[\bigcup \mathcal{Q}]$ has connected components of weight at most $\frac{w}{2}$ throughout the recursive steps. In each step of recursion, we further split the induced path(s) in $\mathcal{Q}$ by putting at most four paths of length at most $t+1$ in $\mathcal{P}$ (i.e., the set of paths in the second outcome of Theorem 2). Hence, we are able to use Theorem 11 to obtain an extended strip decomposition (H, η). If (H, η) is already refined, then we are done. Otherwise, it contains a particle A that is not small. We use Theorem 14 to select at most two paths touching A. By adding two two-vertex paths to $\mathcal{P}$ and deleting their neighborhood, we can separate A (together with the respective touching paths) from the rest of the graph. Then the graph induced by A and the touching paths form a smaller instance, i.e., an instance where $|\bigcup \mathcal{Q}|$ drops by a factor of $\frac{2}{3}$. We ensured that at every recursive step, we included only a constant number of paths of length at most $t+1$ into $\mathcal{P}$. We now prove the core recursive formulation of the algorithm formally.

Lemma 15 (Recursion)**.** *Given a graph G and $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$, integer t, a set $\mathcal{Q}$ of at most two induced paths (vertex disjoint non-adjacent), and a refined rigid extended strip decomposition of $G - N[\bigcup \mathcal{Q}]$. In polynomial time, we can output one of the following:*

- *an induced copy of $S_{t,t,t}$ in G, or*
- *$\mathcal{P}$, $X \subseteq N[\bigcup \mathcal{P}]$, and a refined extended strip decomposition (H, η) of $G - X$, so that $\mathcal{P}$ is a set of induced paths, $|\mathcal{P}| \leq 6 \log_{3/2} (|\bigcup \mathcal{Q}|) + 6$, and the longest path in $\mathcal{P}$ has at most $t + 1$ vertices.*

Proof. If the longest path of $\mathcal{Q}$ has at most $3t + 1$ vertices, return $\mathcal{P} := \mathcal{Q}$ where each path in $\mathcal{P}$ may be further split in at most three paths on at most $t + 1$ vertices, and $X := N[\bigcup \mathcal{P}]$. Hence, we output the extended strip decomposition we were given by the assumptions of the lemma.

Otherwise, let Q_1 be the longest path in $\mathcal{Q}$. Let u_1 and u_2 be the $\left(\left\lfloor \frac{|Q_1|}{3} \right\rfloor + 1\right)$-th and the $\left(2 \left\lfloor \frac{|Q_1|}{3} \right\rfloor + 2\right)$-th vertex of Q_1, respectively. The removal of u_1 and u_2 from Q_1 divides the path into three induced non-touching subpaths Q_1^1, Q_1^2, and Q_1^3, each of length at least t. Let Q_2 be the remaining path of $\mathcal{Q}$, should it exist. We define $\mathcal{S} := \{Q_1^1, Q_1^2, Q_1^3, Q_2\}$ if Q_2 exists, or $\mathcal{S} := \{Q_1^1, Q_1^2, Q_1^3\}$, otherwise. Consult Figure 3.2 to see an overview of the definitions described in this paragraph. For each path $P \in \mathcal{S}$ we define $\mathrm{pref}(P)$ as the set comprising:

- first $t - 1$ vertices of P (or all vertices of P if $|P| < t - 1$), and
- the separating vertex of Q_1 directly preceding P if $P \in \{Q_1^2, Q_1^3\}$.

It can be easily seen that the set of vertices $\mathrm{pref}(P)$ forms an induced path of length at most t. We finally define *shells* of paths in $\mathcal{S}$. Given a path $P \in \mathcal{S}$, we set $\mathrm{shell}(P) := N[\mathrm{pref}(P)] \setminus \bigcup \mathcal{S}$ if $|P| \geq t$ and $\mathrm{shell}(P) := N[\mathrm{pref}(P)]$ otherwise. Intuitively, if $|P| < t$, the shell of P takes the whole neighborhood as we do not have a use for a short path in the next stage of our algorithm. For a long enough path P ($|P| > t$), the shell of P intersects all short paths (shorter than t) connecting the first vertex of P with the rest of the graph. In other words, each path from the first vertex of P to any vertex of $G - \mathrm{shell}(P)$ outside of P will have length at least t. To ease the notation, we define $\mathcal{S}_{\geq t} := \{P \in \mathcal{S} \mid |P| \geq t\}$, $\mathrm{shell}(\mathcal{S}) := \bigcup_{P \in \mathcal{S}} \mathrm{shell}(P)$, and $\mathrm{pref}(\mathcal{S}) := \bigcup_{P \in \mathcal{S}} \mathrm{pref}(P)$.

Now, we use the algorithm from Theorem 11 on Z being the set of the first vertices of paths in $\mathcal{S}_{\geq t}$ and the graph defined as $G - \mathrm{shell}(\mathcal{S})$. If Theorem 11 produces an induced tree containing three elements of Z, G contains an induced $S_{t,t,t}$, since the induced tree must consist of three induced non-touching paths on at least t vertices in $G - \mathrm{shell}(\mathcal{S})$. Hence, we obtained an extended strip decomposition (H', η') of $G - \mathrm{shell}(\mathcal{S})$. If the obtained decomposition is refined, we return $\mathcal{P} := \mathrm{pref}(\mathcal{S})$, $X := \mathrm{shell}(\mathcal{S})$, and the extended strip decomposition $(H := H', \eta := \eta')$.

Therefore, the obtained extended strip decomposition (H', η') of $G - \mathrm{shell}(\mathcal{S})$ contains a particle A which is not small, i.e., $\mathfrak{w}(V(A)) > \frac{w}{2}$. As every vertex in Z is peripheral in (H', η'), we know that no three paths in $\mathcal{S}_{\geq t}$ touch one particle by Theorem 14. Therefore, we take the set $\hat{\mathcal{Q}}$ of at most two paths, say P_1 and P_2, touching A (for convenience, let P_1 or P_2 be an empty set if it does not exist). We now compute the maximum size of $\hat{\mathcal{Q}}$ with respect to $\bigcup \mathcal{Q}$. If both $P_1, P_2 \subseteq Q_1$, then this fraction is at most $\frac{2}{3}$ as by the definition $|Q_1^i| \leq \frac{|Q_1|}{3}$, for $i \in \{1, 2, 3\}$. If one is Q_2 and the other comes from Q_1, then we estimate $a + \frac{1-a}{3} = \frac{2a+1}{3} \leq \frac{2}{3}$ for $a = |Q_2| / |\bigcup \mathcal{Q}| \leq \frac{1}{2}$. Hence, we know that $|\bigcup \hat{\mathcal{Q}}| \leq \frac{2}{3} |\bigcup \mathcal{Q}|$. We define $\hat{G} := A \cup P_1 \cup P_2$ to use Lemma 15 on a smaller instance. Now, we need to verify that the assumption of the lemma holds. We claim the following:

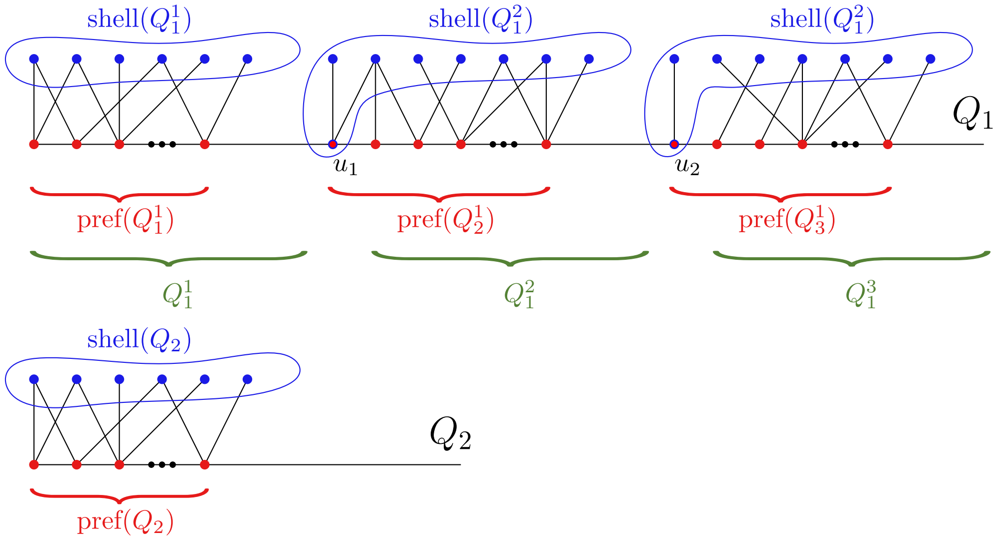


Figure 3.2 Definitions of $\mathrm{pref}(\mathcal{S})$ and $\mathrm{shell}(\mathcal{S})$ in case of $|Q_2| \geq t$.

Claim 16.16. *$\hat{G} - N[\bigcup \hat{\mathcal{Q}}]$ has a refined extended strip decomposition.*

Proof. As $\hat{G}$ is an induced subgraph of G and $G - N[\bigcup \mathcal{Q}]$ has a refined extended strip decomposition, we know that $\hat{G} - N[\bigcup \mathcal{Q}]$ has a refined extended strip decomposition. First, recall that $N[u_1] \setminus (Q_1^1 \cup Q_1^2) \subseteq \mathrm{shell}(Q_1^2)$, which is disjoint with $V(\hat{G})$. Analogously $N[u_2] \setminus (Q_1^2 \cup Q_1^3)$ is disjoint with $V(\hat{G})$. Also, if $|Q_2| < t$ then Q_2 is disjoint with $V(\hat{G})$ as well. Hence, $\hat{G} - N[\bigcup \mathcal{Q}] \simeq \hat{G} - N[\bigcup \mathcal{S}_{\geq t}]$. Also, recall that the only paths among $\mathcal{S}_{\geq t}$ that touch A are in $\hat{\mathcal{Q}}$. Hence, observe that $\hat{G} - N[\bigcup \mathcal{S}_{\geq t}] \simeq \hat{G} - N[\bigcup \hat{\mathcal{Q}}]$. $\square$

Therefore, we can apply Lemma 15 inductively on $\hat{G}$ and $\hat{\mathcal{Q}}$, obtaining $\hat{\mathcal{P}}$ and $\hat{X}$, and a refined extended strip decomposition $(\hat{H}, \hat{\eta})$ of $\hat{G} - \hat{X}$. We need to combine the extended strip decomposition obtained from the recursion with the extended strip decomposition (H', η') we obtained earlier.

We can always suppose that particle A is of type A_{xy}^{xy} for some edge $xy \in E(H')$, unless A is of type A_x for an isolated vertex $x \in V(H')$. That is because A_{xy}^{xy} is the superset of all possible particle types. As Theorem 11 gives us that both $\eta'(xy, x)$ and $\eta'(xy, y)$ are nonempty, we can select $v_x \in \eta'(xy, x)$ and $v_y \in \eta'(xy, y)$ (possibly $v_x = v_y$). By Observation 13, the set

$$X' := (N(v_y) \cup N(v_x)) \setminus V(A)$$

separates A from the rest of G. Set $\mathcal{P}' := \{\{v_x\}, \{v_y\}\}$. In the case of A_x such that $x \in V(H)$ is an isolated vertex, we set $\mathcal{P}' := \emptyset$ and $X' := \emptyset$ and still such A is separated from the rest of G by X'. We return:

- $\mathcal{P} := \hat{\mathcal{P}} \cup \mathcal{P}' \cup \mathrm{pref}(\mathcal{S})$,
- $X := \hat{X} \cup X' \cup \mathrm{shell}(\mathcal{S})$,
- an extended strip decomposition (H, η) of $G - X$, where H is $\hat{H}$ with an additional isolated vertex w, and η is $\hat{\eta}$ restricted only to vertices in $A \setminus X$ with an additional

trivial vertex particle $\eta(w)$ containing all vertices of $G - X - A$. Indeed, $\mathfrak{w}(V(G - X - A)) \leq \frac{w}{2}$ as $\mathfrak{w}(V(A)) > \frac{w}{2}$.

Recall that during the recursion, we do not require rigidity, therefore, we do not mind restricting η only to a subset of vertices. Note that indeed, $G - X - A$ may contain parts of P_1 or P_2, however, $\eta(w)$ does not touch any vertices contained in $\hat{\eta}$ restricted to $A \setminus X$ as $X' \subseteq X$ completely separated $A \setminus X$ from $G - X - A$.

We compute that $|\mathcal{P}| \leq 6 + 6\log_{3/2}\left(\left|\bigcup \hat{\mathcal{Q}}\right|\right) + 6 \leq 6\log_{3/2}(|\bigcup \mathcal{Q}|) + 6$ as we added at most six new paths into $\mathcal{P}$. Observe that the described algorithm runs in polynomial time as we just computed that the depth of recurrence is logarithmic in $|\bigcup(\mathcal{Q})| \leq |V(G)|$ and each recursive call takes polynomial time in the size of G. □

Proof of Theorem 2. Using Theorem 1 we find a weighted Gyárfás path Q. We get the desired outcome by Lemma 15(Recursion) on G with $\mathcal{Q} \coloneqq \{Q\}$. The extended strip decomposition needed by the lemma's assumption is trivial. That is, each connected component of $G - Q$ is represented by a vertex particle of small size. Note that Lemma 15 provides an extended strip decomposition of $G - X$, where $X \subseteq \bigcup \mathcal{P}$ and Theorem 2 only requires an extended strip decomposition of $G - \bigcup \mathcal{P}$, so we can restrict the obtained extended strip decomposition to $V(G) \setminus \bigcup \mathcal{P}$. We conclude the recursion bound of Theorem 2 by the following calculation:

$$6\log_{3/2} n + 6 \leq 11\log n + 6.$$

It only remains to show that we can restore the rigidity assumption by deleting a neighborhood of at most two vertices, which will increase the bound to the final $11\log n + 8$. The plan is as follows. We will modify the given refined extended strip decomposition of G to create a rigid refined extended strip decomposition of G. At first, we ensure that all the edge-interfaces are non-empty (so satisfying rigid). However, after this step, the obtained decomposition do not have to satisfy fully the definition of extended strip decomposition due to triangle sets. Second, we move all the triangle sets that breaks the definition (repairing our decomposition but possibly breaking refiness). Third, we argue, that after this step, we either have a rigid refined extended strip decomposition or it is enough to delete two additional vertices together with their neighborhoods to have all connected components small.

Let (H, η) be a refined extended strip decomposition of G that is not rigid. Observe that we can simply remove any empty trivial vertex particle from η as well as the corresponding isolated vertex from H. Moreover, we can easily modify any extended strip decomposition to add the assumption that sets $\eta(xy, x) \neq \emptyset$ and $\eta(xy, y) \neq \emptyset$ for any edge $xy \in E(H)$. Indeed, if both $\eta(xy, x)$ and $\eta(xy, y)$ are empty then we remove the edge xy from H and put the vertices from $\eta(xy)$ into a newly created isolated vertex of H. If only $\eta(xy, x) = \emptyset$ then we update (H, η) by creating a new vertex x' of degree 1 adjacent only to y in H and setting $\eta(x'y) \coloneqq \eta(xy)$, $\eta(x'y, x') = \eta(x'y, y) \coloneqq \eta(xy, y)$, and removing xy from H. After these modifications, no edge of H has an empty interface. However, as we removed some edges of H, sets $\eta(xyz)$ do not need to correspond to triangles in H anymore.

Let us now discuss how to deal with sets originally corresponding to triangles. If $\eta(xyz)$ is completely disconnected, put its vertices into a new isolated vertex of H. If $\eta(xyz)$ is connected to xy and disconnected to its other edges or they do not exist, we add vertices of $\eta(xyz)$ to $\eta(xy) \setminus (\eta(xy,x) \cup \eta(xy,y))$ (i.e., we put its vertices to the interior of edge xy). Observe that so far all the mentioned modifications did not increase the maximum weight of a particle. Thus, all those reductions will restore rigidity and keep the property of being refined. It remains to consider the last situation where there is a set $\eta(xyz)$ such that it is adjacent to exactly two edges of H, say $\eta(xz)$ and $\eta(zy)$. In this case, we move vertices from $\eta(xyz)$ into $\eta(z)$ (and delete $\eta(xyz)$); note that in this situation, the degree of z in H is at least 2. After all these operations, our decomposition became again a rigid extended strip decomposition, denoted as (H', η'). However, by this operation, we could potentially create a particle that is not small.

Let A be such a particle of (H', η'), i.e., $(\mathfrak{w}(V(A)) > \frac{w}{2})$. Without loss of generality, it is a full-edge particle $A = A^{za}_{za}$, as any vertex particle is included also in a full-edge particle unless it corresponds to an isolated vertex. We take an arbitrary vertex v_z of $\eta'(za, z)$ and an arbitrary vertex v_a of $\eta'(za, a)$, possibly $v_z = v_a$ (they exist by rigidity). Observe that once $N[\{v_z, v_a\}]$ is removed, vertices of A is disconnected from the rest of the graph, which has a small weight. Moreover, $\eta'(z)$ and $\eta'(a)$ consist of some of the triangles of the original extended strip decomposition (H, η) that were added during the modification process and the original vertex ηz and ηa, respectively. Thus, after the removal of $N[\{v_z, v_a\}]$, all these triangles are either disconnected or were part of the original full-edge particle for the edge az of H, but (H, η) was refined. Therefore, all connected components of $G \setminus N[\{v_z, v_a\}]$ are small. $\square$

The following simple corollary is a generalization of Theorem 2 that is useful for $sS_{t,t,t}$-free graphs, for some $s, t \geq 1$.

Corollary 17. *Given an n-vertex graph G with nonnegative vertex weights and $s, t \geq 1$, one can in polynomial time either:*

- *output an induced copy of $sS_{t,t,t}$ in G, or*
- *output a set X consisting of at most*

$$(s-1)(3t+1) + (11 \log n + 8)(t+1)$$

vertices and a rigid extended strip decomposition of $G - N[X]$ whose every particle has weight at most half of the total weight of $V(G)$.

Proof. Note that we can safely assume that $s \leq n$, as otherwise we can return $X = V(G)$ as the second outcome. Induction on s. If $s = 1$, then we obtain the result immediately by Theorem 2. Thus let us assume that $s \geq 2$ and the theorem holds for $s - 1$.

We apply Theorem 2 to G and t. If the algorithm returns its second outcome, i.e., a set $\bigcup \mathcal{P}$ of size at most $(11 \log n + 8)(t+1)$ and an extended strip decomposition of $G - N[\bigcup \mathcal{P}]$, we return this as the second outcome as well (with $X = \bigcup \mathcal{P}$).

So suppose that the algorithm returned a some $Y \subseteq V(G)$ with $|Y| = 3t + 1$, such that $G[Y] \simeq S_{t,t,t}$. We apply induction on $G' := G - N[Y]$, $s - 1$, and t. Let w' (w) be the weight of G' (G).

If the inductive call returns an induced $(s-1)S_{t,t,t}$ in G', then, together with Y, we obtain an induced $sS_{t,t,t}$ in G and we return it as the first outcome. In the other case, the inductive call returns a set $X' \subseteq V(G')$ of size at most $(s-2)(3t+1) + (11 \log n' + 6)(t+1)$

and a rigid extended strip decomposition (H, η) of $G' - N[X']$ whose every particle has weight at most $w'/2$. We set $X = Y \cup X'$. Now X and (H, η) satisfy the statement of the theorem, as $G' - N[X'] = G - N[X]$ and $w' \leq w$. The total running time is polynomial in n as the depth of the recursion is $s - 1 \leq n$. □

3.3 Algorithmic Applications

In this section, we will show how to combine Theorem 2 with the approach of Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [53] in order to obtain a QPTAS and a subexponential-time algorithm for MWIS in $sS_{t,t,t}$-free graphs, i.e., we prove Theorems 3 and 4.

Both algorithms follow the same general outline; let us sketch it before we get into the details of each particular case. Each algorithm is a recursive procedure, which consists of two phases. In the first one, we deal with the vertices of G that are *heavy*, which means that their neighborhood is "large", where the exact meaning of "large" depends on the particular algorithm.

Once there are no heavy vertices, i.e., the neighborhood of each vertex is "small", we proceed to the second phase. We call Corollary 17 for the current instance G, obtaining a small-sized set X and a rigid extended strip decomposition (H, η) of $G - N[X]$, whose every particle is of small size. The crux is that since we are in the second phase, all vertices in X are not heavy, and since X is of small size, the whole set $N[X]$ is "small". We treat $N[X]$ separately in a way that depends on the particular algorithm.

Next, for each particle A of (H, η), we call the algorithm recursively for $G[A]$, obtaining (a good approximation of) a maximum-weight independent set in $G[A]$. Finally, we combine the obtained results to derive (a good approximation of) a maximum-weight independent set in G. This last step is based on the idea of Chudnovsky et al. [53] to reduce the problem of finding a maximum-weight matching in a graph obtained by a simple modification of H. Since the size of H is linear in $|V(G)|$ (by Observation 12), this problem can be solved in time polynomial in $|V(G)|$ using, e.g., the classic algorithm of Edmonds [82]. The last step is encapsulated in the following lemma, whose exact statement comes from Abrishami, Chudnovsky, Dibek, and Rzążewski [1].

Lemma 18 (Chudnovsky et al. [53]). *Let $\varsigma \in [0, 1]$ be a real number. Let G be an n-vertex graph equipped with a weight function $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$. Suppose that G is given along with an extended strip decomposition (H, η), where H has N vertices.*
Let $I_0 \subseteq V(G)$ be a fixed independent set in G. Furthermore, assume that for each particle A of (H, η) we are given an independent set $I(A)$ in $G[A]$ such that $\mathfrak{w}(I(A)) \geq \varsigma \cdot \mathfrak{w}(I_0 \cap A)$. Then in time polynomial in $n + N$ we can compute an independent set I in G such that $\mathfrak{w}(I) \geq \varsigma \cdot \mathfrak{w}(I_0)$.

Let us stress out that the algorithm from Lemma 18 does not need to know the value of ς or the independent set I_0.

The main difference between our approach and the one of Chudnovsky et al. [52] is that we use Theorem 2 and its consequence, i.e., Corollary 17. The previous algorithms used a similar statement but with a worse (and much more involved) guarantee on the size of X and each particle. Furthermore, the way we obtain our set X is significantly simpler.

In the proofs of Theorems 3 and 4, we use Corollary 17 with the uniform weights only. That means, the rigid extended strip decomposition given by Corollary 17 has every particle of size at most half of the number of vertices in the original graph.

3.3.1 Proof of Theorem 3

We restate the theorem here:

Theorem 3. *Given an n-vertex graph G with weights on vertices and integers $s, t \geq 1$, one can in time exponential in $\mathcal{O}(\sqrt{stn}\log n)$ output one of the following outcomes:*
1. an induced $sS_{t,t,t}$ in G, or
2. an independent set in G of maximum possible weight.

Before we proceed to the proof, let us first explain the meaning of "small", and how to deal with $N[X]$ in this particular case. Here the neighborhood of a vertex is "small" if it has few vertices (more specifically, at most $\sqrt{n/t}$). In the first phase, we deal with heavy vertices v (i.e., of large degree) with simple branching: we guess whether v is included in our optimum solution or not. Since the degree of v is large, in the first branch, we obtain significant progress, which is enough to obtain a subexponential running time.

In the second phase, since $N[X]$ is the neighborhood of $\mathcal{O}(\log n)$ vertices, each of degree $\mathcal{O}(\sqrt{n})$, the total size of $N[X]$ is $\mathcal{O}(\sqrt{n}\log n)$. Thus we can afford to exhaustively guess the intersection of our optimum solution with $N[X]$.

Proof of Theorem 3. Let $s, t \geq 1$ be integers and let $(G, \mathfrak{w})$ be an instance of MWIS, where G has n vertices. We observe that if n is small, i.e., bounded by a constant, then we can solve the problem by brute force. Thus we assume that $n \geq n_0$, where n_0 is a constant whose exact value follows from the reasoning below.

First, consider the case that there exists $v \in V(G)$ such that $\deg v \geq \sqrt{n/st}$. We branch on including v in the final solution: we either delete v from G, or we delete $N[v]$ and add v to the solution returned by the recursive call. Then we output the one of these two solutions that has a larger weight. The correctness of this step of the algorithm is straightforward.

Hence, we can assume that for every $v \in V(G)$ it holds that $\deg v \leq \sqrt{n/st}$. We call Corollary 17. If the algorithm finds an induced $sS_{t,t,t}$, we return it as the first outcome. In the other case, we obtain a set X of size $(s-1)(3t+1)+(11\log n+8)(t+1) \leq 12st\log n$ (here we use that n is large), and a rigid extended strip decomposition (H, η) of $G' = G - N[X]$ whose every particle has at most $n/2$ vertices.

We exhaustively guess an independent set $J \subseteq N[X]$; think of it as an intersection of the intended optimum solution with $N[X]$. Consider the graph $G'' := G' - N[J]$. We modify (H, η) by removing the vertices from $N[J]$ from the sets $\eta(\cdot)$. Let us call the obtained strip decomposition (H, η'); note that it might not be rigid. We call the algorithm recursively for the subgraph $G''[A]$ for every nonempty particle A of (H, η'). Let $I(A)$ be the solution. If $A = \emptyset$, then $I(A) = \emptyset$. By the inductive assumption $I(A)$ is a maximum-weight independent set in $G''[A]$. Then we use Lemma 18 for $\varsigma = 1$ to combine the solutions into a maximum-weight independent set I_J of G''. Finally, we return the independent set $J \cup I_J$ whose weight is maximum over all choices of J. Note that the correctness of this step is guaranteed by the exhaustive guessing of J and Lemma 18.

Running time. Let $F(n)$ denote the running time of our algorithm for n-vertex instances. We prove that $F(n) = 2^{\mathcal{O}\left(\sqrt{stn}\log n\right)}$. If $n < n_0$, then the claim clearly holds. So let us assume that $n \geq n_0$.

In the first case, we call the algorithm for two instances, one of size $n-1$ and one of size at most $n - \sqrt{n/st}$. Hence,

$$F(n) \leq F(n-1) + F(n - \sqrt{n/st}) = 2^{\mathcal{O}\left(n\log n/\sqrt{\frac{n}{st}}\right)} \leq 2^{\mathcal{O}\left(\sqrt{stn}\log n\right)}.$$

Here we skip the description how this recursion is solved, as it is pretty standard. For a formal proof we refer the reader to Bacsó, Lokshtanov, Marx, Pilipczuk, Tuza, and van Leeuwen [11, Lemma 1].

It remains to analyze the running time of the step in which the maximum degree of vertices in G is bounded by $\sqrt{n/st}$. Corollary 17 asserts that we obtain X and the rigid extended strip decomposition (H, η) of $G' = G \setminus N[X]$ in time polynomial in n. There are $2^{\mathcal{O}(\sqrt{n/st}\cdot st\log n)} = 2^{\mathcal{O}(\sqrt{stn}\log n)}$ ways of choosing the set J. In polynomial time we modify (H, η) into (H, η').

Observe that while (H, η') might not be rigid, it was obtained from a rigid extended strip decomposition (H, η) by deleting some vertices from the sets $\eta(\cdot)$. In particular, both decompositions have the same sets of particles, and every nonempty particle of (H, η') is also a nonempty particle of (H, η). Thus by Observation 12 we call the algorithm recursively for at most $4n$ nonempty particles, each of size at most $n/2$. Finally, having computed a maximum-weight independent set contained in each particle, by Lemma 18, we can compute the final solution in time polynomial in n. Hence, there are constants c, c_1, c_2, where $c \gg c_1, c_2$, such that total running time of this step is bounded by:

$$F(n) \leq 2^{c_1\cdot\sqrt{stn}\log n}\left(n^{c_2} + 4n \cdot 2^{c\cdot\sqrt{stn/2}\log(n/2)}\right) \overset{c\gg c_1,c_2}{\leq} 2^{c\cdot\sqrt{stn}\log n}, \tag{3.1}$$

and so is the total complexity of the algorithm. □

3.3.2 Proof of Theorem 4

For the convenience, we restate the theorem here.

Theorem 4. *Given an n-vertex graph G with weights on vertices, integers $s, t \geq 1$, and a real $\varepsilon > 0$, one can in time exponential in $\mathcal{O}(\varepsilon^{-1}st\log^5 n)$ output one of the following outcomes:*

1. *an induced $sS_{t,t,t}$ in G, or*
2. *an independent set in G that is within a factor of $(1-\varepsilon)$ of the maximum possible weight.*

Again let us start with explaining the algorithm-specific details of the outline presented at the start of Section 3.3.

We will use the notion of β-heavy vertices from [53]. Consider a graph G, a weight function $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$, and an independent set $I \subseteq V(G)$. Let $\beta \in (0, 1/2]$ be a real. We say that a vertex $v \in V(G)$ is *β-heavy* (with respect to I) if $\mathfrak{w}(N[v] \cap I) > \beta \cdot \mathfrak{w}(I)$. A set J is *good for I* if $J \subseteq I$ and $N[J]$ contains all vertices that are β-heavy with respect to I.

Lemma 19 (Chudnovsky et al. [53]). *Let G be an n-vertex graph for $n > 2$, $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$ be a weight function, $I \subseteq V(G)$ be an independent set, and $\beta \in (0, 1/2]$ be a real. Then there exists a set J of size at most $\lceil \beta^{-1} \log n \rceil$ which is good for I.*

Now the vertex is heavy if it is β-heavy for some carefully chosen parameter β. This means that a neighborhood of a vertex is "large" if it contains a significant ($\geq \beta$) fraction of the weight of I_{OPT}. In the first phase, we exhaustively guess the set J that is good for a fixed optimum solution I_{OPT}. Note that J is of small size and since $J \subseteq I_{\mathsf{OPT}}$, we know that $N(J)$ contains no vertices from I_{OPT} and thus can be safely removed from the graph.

Since J is good for I_{OPT}, we know that $G - N[J]$ contains no heavy vertices, and for this graph we call Corollary 17. Now, as $N[X]$ is a neighborhood of few non-heavy vertices, we know that the total weight of $I_{\mathsf{OPT}} \cap N[X]$ is small and thus can be sacrificed, as we aim for an approximation.

Proof of Theorem 4. Let $s, t \geq 1$ be inetegrs and let $(G, \mathfrak{w})$ be an instance of MWIS, where G has n vertices. Let $\varepsilon \in (0, 1)$ be fixed. Fix a maximum-weight independent set I_{OPT} in G with respect to $\mathfrak{w}$. We describe a procedure that either finds in G an induced $sS_{t,t,t}$ or an independent set I of weight at least $(1 - \varepsilon) \cdot \mathfrak{w}(I_{\mathsf{OPT}})$.

Let N be the minimum power of two greater than or equal to the size of our initial instance. Note that $n \leq N < 2n$. The value of N will not change throughout the execution of the algorithm.

The algorithm itself is a recursive procedure. The arguments of each call are a graph G', which is an induced subgraph of G, the weight function on $V(G')$ obtained by restricting the domain of $\mathfrak{w}$, and an integer h, which can be intuitively understood as the depth of the current call in the recursion tree. Since it does not lead to confusion, we will always denote the weight function by $\mathfrak{w}$. We will keep the invariant that for each call $(G', \mathfrak{w}, h)$ it holds that $|V(G')| \leq N/2^h$. The initial call, corresponding to the root of the recursion tree, is for $(G, \mathfrak{w}, 0)$.

Consider a call for the instance $(G', \mathfrak{w}, h)$. If $|V(G')| < n_0$, where n_0 is a constant that follows from the reasoning below, then we can solve the problem by brute force. Thus let us assume that $n \geq n_0$. In particular, $N > 1$.

We set

$$\beta(h, \varepsilon) := \frac{\varepsilon}{12st \log (N/2^h) \cdot ((1 - \varepsilon) \log N + \varepsilon(h + 1))}. \tag{3.2}$$

It is straightforward to verify that for $h < \log N$ we have $\beta(h, \varepsilon) \in (0, 1/2]$. On the other hand, if $h \geq \log N$, then G' is of constant size and thus $\beta(h, \varepsilon)$ is not computed for such h.

Let $\mathcal{J}$ be the family of all independent sets in G' of size at most $\lceil \beta(h, \varepsilon)^{-1} \log (N/2^h) \rceil$. For each $J \in \mathcal{J}$ we proceed as follows. If $|V(G' - N[J])| < n_0$, then we compute a maximum-weight independent set I_J in $G' - N[J]$ by brute force. Otherwise, we use Corollary 17, to obtain a set $X_J \subseteq V(G' - N[J])$ and a rigid extended strip decomposition (H, η) of $G' - N[J] - N[X_J]$ such that each particle of (H, η) is of size at most $|V(G' - N[J])|/2$. By Corollary 17 and since n is large, we obtain

$$\begin{aligned} |X_J| &\leq (s - 1)(3t + 1) + (11 \log |V(G' - N[J])| + 8)(t + 1) \\ &\leq 12st \log |V(G')| \leq 12st \log(N/2^h). \end{aligned} \tag{3.3}$$

Let $Y_J \coloneqq N(J) \cup N[X_J]$. We modify (H, η) into an extended strip decomposition of $G' - Y_J$ as follows. For each $v \in J$, we add to H an isolated vertex x_v, and set $\eta(x_v) = \{v\}$.[1] Let us call this extended strip decomposition (H', η'). Observe that each particle of (H', η') is of size at most $|V(G' - N[J])|/2 \leq |V(G')|/2$. Furthermore, since (H, η) is rigid, so is (H', η').

For each nonempty particle A of (H', η') we call the algorithm recursively on an instance $(G'[A], \mathfrak{w}, h+1)$. Let $I(A)$ be the value returned by the algorithm. For each empty particle A, we set $I(A) \coloneqq \emptyset$. Finally, we apply the algorithm from Lemma 18, in order to obtain an independent set I_J of $G' - Y_J$ and thus of G'. Recall that the value of ς is not needed to apply Lemma 18; we will define it in the next paragraph when we discuss the approximation guarantee. As the solution, we return the set I_J of maximum weight, over all choices of $J \in \mathcal{J}$.

Approximation guarantee. Consider the recursion tree of our algorithm. We mark some nodes of the recursion tree. First, we mark the root. Now consider some marked node z corresponding to a call $(G', \mathfrak{w}, h)$, such that z is not a leaf node. Observe that by Lemma 19, there is some $J \in \mathcal{J}$ (for this particular instance) which is good for $I_{\mathsf{OPT}} \cap V(G')$. Fix such J. If there is more than one, we choose one arbitrarily. We mark the children of z that correspond to the calls on the particles of the extended strip decomposition of $G' - Y_J$.

Let $\mathcal{T}$ be the subtree of the recursion tree induced by the marked nodes. Note that each leaf of $\mathcal{T}$ is a leaf of the whole recursion tree, i.e., it corresponds to an instance of constant size. Since at each level of the recursion, the size of the instance drops by at least half, we observe that each instance at level h (where the root is at level 0) is of size at most $N/2^h$. Consequently, the depth of $\mathcal{T}$ is at most $\log N$.

Consider a call for an instance $(G', \mathfrak{w}, h)$ and let J be good for I_{OPT}. Let us estimate $\mathfrak{w}(I_{\mathsf{OPT}} \cap Y_J)$. First, observe that since $J \subseteq I_{\mathsf{OPT}}$, we have that $\mathfrak{w}(I_{\mathsf{OPT}} \cap N(J)) = 0$. Moreover, since J was chosen to be good, there are no $\beta(h, \varepsilon)$-heavy vertices in $V(G' - N[J])$, and in particular, in $N[X_J]$. Hence,

$$\begin{aligned} \mathfrak{w}(I_{\mathsf{OPT}} \cap Y_J) &= \mathfrak{w}(I_{\mathsf{OPT}} \cap N[X_J]) \leq |X_J| \cdot \beta(h, \varepsilon) \cdot \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G')) \\ &\overset{(3.2) \text{ and } (3.3)}{\leq} \frac{\varepsilon}{(1-\varepsilon)\log N + \varepsilon(h+1)} \cdot \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G')). \end{aligned} \tag{3.4}$$

The following claim shows that the solution computed for the instance $(G', \mathfrak{w}, h)$ at each node of $\mathcal{T}$ is a reasonable approximation of $I_{\mathsf{OPT}} \cap V(G')$.

Claim 20.20. *Let z be a node of $\mathcal{T}$, and let $(G', \mathfrak{w}, h)$ be the instance corresponding to z. Let I be the independent set returned by the algorithm for the call at z. Then $\mathfrak{w}(I) \geq \left(1 - \varepsilon + \frac{\varepsilon h}{\log N}\right) \cdot \mathfrak{w}\left(I_{\mathsf{OPT}} \cap V(G')\right)$.*

Proof. First, observe that if z is a leaf of $\mathcal{T}$, then the statement of the claim is satisfied. Indeed, in this case I is computed by brute force, and hence $\mathfrak{w}(I) = \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G'))$.

[1]Another possible way of dealing with the set J would be to add it directly in the computed solution. However, we decided to restore J to the graph, so that these vertices are handled by Lemma 18 and do not require any special treatment.

Recall that the algorithm returns the solution of maximum weight among all choices of $J \in \mathcal{J}$, so clearly we have $\mathfrak{w}(I) \geq \mathfrak{w}(I_J)$, where J is good for $I_{\mathsf{OPT}} \cap V(G')$.

We proceed by induction on h. First, consider a node z at the level $h = \log N$. As the depth of $\mathcal{T}$ is at most $\log N$, we observe that z must be a leaf, so the claim follows by the observation above.

Assume that the claim holds for $h+1 \in [\log N]$ and consider a node z at level h. If z is a leaf, then again, we are done. Otherwise, let $\mathcal{A}$ be the set of nonempty particles of the extended strip decomposition of $G' - Y_J$. For every such particle A, we recursively computed an independent set $I(A)$. By the inductive assumption, we have that $\mathfrak{w}(I(A)) \geq \left(1 - \varepsilon + \frac{\varepsilon(h+1)}{\log N}\right) \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G'[A]))$; note that these recursive calls are at level $h+1$. Clearly, the same holds for empty particles because $\emptyset$ is there an optimum solution.

Thus, by Lemma 18 applied to I_{OPT} and $\varsigma = 1 - \varepsilon + \frac{\varepsilon(h+1)}{\log N}$, we obtain an independent set I_J in $G' - Y_J$, such that

$$\begin{aligned} \mathfrak{w}(I_J) \geq & \left(1 - \varepsilon + \frac{\varepsilon(h+1)}{\log N}\right) \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G' - Y_J)) \\ = & \left(1 - \varepsilon + \frac{\varepsilon(h+1)}{\log N}\right) \Big(\mathfrak{w}(I_{\mathsf{OPT}} \cap V(G')) - \mathfrak{w}(I_{\mathsf{OPT}} \cap Y_J)\Big). \end{aligned} \tag{3.5}$$

Combining (3.5) with (3.4) and simplifying the formula, we obtain

$$\mathfrak{w}(I_J) \geq \left(1 - \varepsilon + \frac{\varepsilon h}{\log N}\right) \mathfrak{w}(I_{\mathsf{OPT}} \cap V(G')),$$

which concludes the proof of the claim. □

Since the root of the recursion tree belongs to $\mathcal{T}$, the final result I returned for the call at the root (i.e., for $(G, \mathfrak{w}, 0)$) satisfies

$$\mathfrak{w}(I) \geq (1 - \varepsilon) \cdot \mathfrak{w}\left(I_{\mathsf{OPT}} \cap V(G)\right) = (1 - \varepsilon) \cdot \mathfrak{w}(I_{\mathsf{OPT}}).$$

This concludes the discussion of the approximation guarantee.

Running time. Recall that the recursion tree has depth at most $\log N$. Let us show the following claim concerning the running time.

Claim 21.21. *Let z be a node of the recursion tree, and let $(G', \mathfrak{w}, h)$ be the instance corresponding to z. Then the algorithm solves this instance in time $2^{\mathcal{O}\left(\varepsilon^{-1} st \log^4 N \log\left(N/2^{h-1}\right)\right)}$.*

Proof. Let $F(h)$ denote the upper bound for the running time of our algorithm, depending on the level of the call in the recursion tree. We aim to show that there is an absolute constant c, such that for N sufficiently large we have

$$F(h) \leq 2^{c \cdot \varepsilon^{-1} st \log^4 N \log\left(N/2^{h-1}\right)}.$$

Recall that $|V(G')| \leq N/2^h$. If z is a leaf, then the instance is of constant size, and thus the claim holds (assuming that c is sufficiently large). In particular this happens if $h = \log N$. So let us assume that the claim holds for the calls at level $h+1$ and that $h < \log N$.

Recall that we first enumerate the family $\mathcal{J}$ of all independent sets of size at most $\lceil \beta(h,\varepsilon)^{-1} \log(N/2^h) \rceil$. Observe that

$$|\mathcal{J}| \leq |V(G')|^{\lceil \beta(h,\varepsilon)^{-1} \log(N/2^h) \rceil} \leq 2^{\log(N/2^h) \lceil \beta(h,\varepsilon)^{-1} \log(N/2^h) \rceil},$$

and the family $\mathcal{J}$ can be enumerated in time polynomial in its size.

For each $J \in \mathcal{J}$, using Corollary 17 and modifying its outcome, in polynomial time we obtain a set X_J and a rigid extended strip decomposition (H', η') of $G - Y_J$, where $Y_J = N[X_J] \cup N(J)$.

Next, we call the algorithm recursively for at most $4 \cdot |V(G')| \leq 4 \cdot N/2^h$ instances, each at depth $h+1$. Finally, use use Lemma 18 to obtain our solution in time polynomial in $|V(G')|$ and thus in $N/2^h$.

Thus the running time is bounded by the following expression (here c_1, c_2, c_3 are absolute constants, such that c_1 and c_2 are much smaller than c_3, and $c_3 = c/12$):

$$\begin{aligned}
F(h) &\leq 2^{c_1 \cdot \beta(h,\varepsilon)^{-1} \log^2(N/2^h)} \cdot \Big((N/2^h)^{c_2} + 4 \cdot (N/2^h) \cdot F(h+1) \Big) \\
&\overset{c_3 \gg c_1, c_2}{\leq} 2^{c_3 \cdot \beta(h,\varepsilon)^{-1} \log^2(N/2^h)} \cdot 2^{c \cdot \varepsilon^{-1} st \log^4 N \log(N/2^h)} \\
&= \exp\left\{ c_3 \cdot \beta(h,\varepsilon)^{-1} \log^2(N/2^h) + c \cdot \varepsilon^{-1} st \log^4 N \log(N/2^h) \right\} \\
&\leq \exp\left\{ c_3 \cdot 12st \cdot \left(\frac{1-\varepsilon}{\varepsilon} \log N + (h+1) \right) \log^3(N/2^h) + c \cdot \varepsilon^{-1} st \log^4 N \log(\frac{N}{2^h}) \right\} \\
&\overset{h < \log N}{\leq} \exp\left\{ c \cdot \varepsilon^{-1} st \log^4 N + c \cdot \varepsilon^{-1} st \log^4 N \log(N/2^h) \right\} \\
&= \exp\left\{ c \cdot \varepsilon^{-1} st \log^4 N (\log(N/2^h) + 1) \right\} = \exp\left\{ c \cdot \varepsilon^{-1} st \log^4 N \log(N/2^{h-1}) \right\}.
\end{aligned}$$

This completes the proof of the claim. □

Now we apply Claim 21.21 to the initial call $(G, \mathfrak{w}, 0)$ and obtain that the overall running time is

$$2^{\mathcal{O}(\varepsilon^{-1} st \log^5 N)} = 2^{\mathcal{O}(\varepsilon^{-1} st \log^5 n)},$$

as $N < 2n$. This completes the proof. □

3.4 Conclusion

We note that compared to the published paper [150], the Chapter 3 of the thesis contains small corrections, including Observation 12 and the way how to preserve the rigidness of a refined extended strip decomposition.

In the QPTAS of Chudnovsky, Pilipczuk, Pilipczuk, and Thomassé [53] it was more convenient to measure the weight of parts of the graph not by the number of vertices, but by the weight of the intersection of the sought solution with the part in question. We observe that we can adapt Theorem 2 to this setting of unknown weight function.

Theorem 22. *Given an n-vertex graph G and an integer t, one can in time $n^{\mathcal{O}(t \log n)}$ either:*

- *output an induced copy of $S_{t,t,t}$ in G, or*
- *output a family $\mathcal{F}$ satisfying the following:*
  1. *every element of $\mathcal{F}$ is a pair of a set $\mathcal{P}$ consisting of at most $11 \log n + 6$ induced paths in G, each of length at most $t+1$, and an extended strip decomposition of $G - N[\bigcup \mathcal{P}]$;*
  2. *for every weight function $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$ there exists a pair in $\mathcal{F}$ such that every particle in the extended strip decomposition of the pair has weight at most half of the total weight of G;*
  3. *the size of $\mathcal{F}$ is bounded by $n^{\mathcal{O}(\log n)}$.*

Proof sketch. As observed in [53], in G one can identify at most n^2 induced paths such that for every weight function $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geq 0}$, at least one of the identified path is a Gyárfás' path for $\mathfrak{w}$, that is, a path Q such that every connected component of $G - N[Q]$ is of weight at most half of the weight of G. Thus, we can guess the path Q as in the proof in Theorem 2 out of at most n^2 candidates.

Then, in the recursive step in the proof of Theorem 2, instead of choosing the heavy particle to recurse on, we guess which particle is heavy (or that none exists). It is easy to see that any extended strip decomposition in the process will have fewer than n inclusion-wise maximal particles; thus, this gives $n^{\mathcal{O}(\log n)}$ possible outputs to enumerate. □

Note that combining Theorem 22 with the approach from Corollary 17, one can obtain an analogous statement when one of the outputs is an induced $sS_{t,t,t}$.

An improvement of our structural result. As we conjectured, it turned out that a deletion of the neighborhoods of only $3t + \mathcal{O}(1)$ vertices is sufficient to obtain a similar statement to Theorem 2 and consequently to Corollary 17 and Theorem 22, as shown recently by Bourneuf, Masaříková, Nadara, and Pilipczuk [24]. The authors started similarly with the Gyárfás' path refining the selection of splitting vertices to query the three-in-a-tree theorem. The main difference lies in a more involved choice of the vertices in Z with a better understanding of the structure of the extended strip decomposition combined with the obvious but powerful property of the two last vertices of the Gyárfás' path. After removing neighborhoods of the Gyárfás' path up to its second-to-last vertex, the largest connected component is still "big" (has a weight bigger than half of the weight of G). While including the single last vertex suddenly drops its size below this threshold. The authors used this property to look more closely at the interaction of the "big" particle and the "big" connected component (where "big" means having more than half of the total weight of the graph).

Let us now describe the key proof idea in greater detail here as we find it helpful for the reader to see the recent improvements and simplifications of the proof presented in the thesis. Let Q be a Gyárfás' path of sufficiently large length. First, to query the three-in-a-tree theorem the following three vertices of Q are taken into consideration instead of splitting Q in thirds: The first vertex of Q (call it x) and the vertices in the distance $t+4$ (call it y) and $t+2$ (call it z) from the end of Q. Then the following three subpaths of length $t+1$ of Q serve as the segments whose neighborhoods outside Q is removed from the graph before querying the three-in-a-tree theorem: the subpaths of Q

starting at x and z and taking the next t vertices in Q, respectively, and the subpath taking the t vertices of Q preceding y. Moreover, the vertex of Q between x and z is removed. Observe that similarly as in the proof of Theorem 2, this selection ensures that the three-in-a-tree theorem returns an extended strip decomposition of the graph without the mentioned neighborhoods.

Before looking at this particular setting, the authors first show a property of any extended strip with respect to an induced path traversing between peripheral vertices.

Lemma 23 ([24])**.** *Whenever P is an induced path in G with endpoints a and b and (H, η) is an extended strip decomposition with a and b being peripheral in it, then $V(P) \subseteq \bigcup_{rs \in E(H)} \eta(rs)$. Moreover, whenever there is $rs \in E(H)$ such that $V(P) \cap \eta(rs) \neq \emptyset$ then $P \cap \eta(rs)$ is a path with exactly one vertex in each of $\eta(rs, r)$ and $\eta(rs, s)$.*

Let us have a closer look at the returned extended strip decomposition and at one of its big particles A (if it does not exist, the returned extended strip decomposition is already an aimed one). We describe now only the most difficult case where A is a full-edge particle, i.e., $A = A_{pq}^{pq}$, but the reader can easily figure out the rest. The authors distinguish the two cases of whether the subpath from x to y in Q (call it Q_1) traverses through A or not. Denote by Q_2 the subpath of Q from z to the end of Q.

If Q_1 is disjoint of A, all the vertices of Q_1 that can have a neighbor in A are the vertices of some of the interfaces of neighboring edges in H, thus, observe there are at most four of them by the definition of extended strip decomposition. Therefore, additional removal of the neighborhoods of these four vertices together with $N[Q_2]$ suddenly drops the weight of the heaviest connected component to at most half of the total weight as Q is Gyárfás' path.

In the second case: Q_1 traverses via A. Thus, by Lemma 23 and the definition of extended strip decomposition, Q_2 does not have any neighbor in A, and vertices of A can be separated from the rest of the graph by four vertices, furthermore, we can choose all of them from Q_1. From the property of Gyárfás' path, $G \setminus N[Q - \ell]$, where ℓ is the last vertex of Q, has a big connected component C but $G \setminus N[Q]$ not, thus, ℓ has a neighbor in C. However, as A is separated by $N[Q_1]$ from the rest of the graph, C and A are disjoint, which is impossible.

Applications of our structural result. Abrishami, Chudnovsky, Dibek, and Rzążewski [2] showed a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs of bounded degree. Their argument is quite involved and revisits the proof of the three-in-a-tree theorem [56]. Very recently, Abrishami, Chudnovsky, Pilipczuk, and Rzążewski [5] showed how to use our main result to obtain the same result as [2] in an arguably simpler way, and with a better running time. The improvement [24] implies the same result almost immediately. Indeed, one needs to branch on $N[P]$ and recurse on the remainder of every particle of (H, η). The maximum degree of H is bounded by a function of the maximum degree of G (i.e., is a constant), which ensures that the sum of sizes of all particles is linear in $|V(G)|$. This in turns implies that the total complexity of the algorithm can be bounded by a polynomial function. Note that the same approach using Theorem 2 yields quasipolynomial running time bound; to get polynomial-time running bound, [5] introduces a "bordered" version of the MWIS problem where the input graph is additionally equipped with $\mathcal{O}(\log n)$ terminals and the task is to compute an independent set of maximum weight for every possible intersection of the solution with the terminals.

We see Theorem 2 as the analog of Theorem 1 in the classes of $S_{t,t,t}$-free graphs: with its help, obtaining a QPTAS or a subexponential algorithm was relatively simple, following the ideas of [11, 51, 53].

Furthermore, very recently Gartland, Lokshtanov, Masařík, Pilipczuk, Pilipczuk, and Rzążewski [99] announced a quasipolynomial-time algorithm for MWIS in $sS_{t,t,t}$-free graphs. Our main result turned out to be the first step of their approach, similarly as Theorem 1 is an essential ingredient of the algorithms for P_t-free graphs [97, 159].

4 On 3-Coloring of $(2P_4, C_5)$-Free Graphs

4.1 Introduction

We study the class of $(2P_4, C_5)$-free graphs. Let us discuss the motivation behind forbidding an induced cycle in addition to $2P_4$. Algorithms for subclasses of P_t-free graphs, which avoid one or more additional induced subgraphs, usually cycles, have been intensively studied. They might be a first step in the attempt to settle the case of P_t-free graphs. This turned out to be the case for 3-coloring of P_7-free graphs (as can be seen from preprints [21, 49, 50] leading to [20]) and 4-coloring of P_6-free graphs [48].

Note that the problem of 4-coloring is NP-complete even when some (P_t, C_ℓ)-free graphs are considered when $t \geq 7$. Hell and Huang [119] and Huang et al. [126] settled many NP-complete cases of this type. These results, in combination with the polynomiality of P_6-free case, leave open only the following cases: (P_7, C_7)-free, (P_8, C_7)-free, and (P_t, C_3)-free graphs, for $7 \leq t \leq 21$.

Chudnovsky and Stacho [60] studied the problem of 3-coloring of P_8-free graphs which additionally avoid induced cycles of two distinct lengths; specifically, they consider graphs that are (P_8, C_3, C_4)-free, (P_8, C_3, C_5)-free, and (P_8, C_4, C_5)-free. For the first two cases, they show that all such graphs are 3-colorable. For the last one, they provide a complete list of 3-*critical graphs*, i.e., the graphs with no 3-coloring such that all their proper induced subgraphs are 3-colorable. Independently, using a computer search, Goedgebeur and Schaudt [103] showed that there are only finitely many 3-critical (P_8, C_4)-free graphs. In fact, 3-coloring is polynomial-time solvable on (P_t, C_4)-free graphs for any $t \geq 1$ [106].

Recently, Rojas and Stein [10] approached the problem by showing that for any odd $t \geq 9$, there exists a polynomial-time algorithm that solves the 3-coloring problem in P_t-free graphs of odd girth at least $t-2$. In particular, their result implies that 3-coloring is polynomial-time solvable for (P_9, C_3, C_5)-free graphs.

Freshly, a similar question was resolved in the case where, instead of a cycle, a 1-subdivision of $K_{1,s}$ (a star with s leaves), denoted as $SDK_{1,s}$, is considered. Chudnovsky, Spirkl, and Zhong have shown that the class of $(SDK_{1,s}, P_t)$-free graphs is list-3-colorable in polynomial time for any $s, t \geq 1$ [57].

Recall that 3-Coloring is open for H-free graphs where $|V(H)| = 8$ only for H being $2P_4$ or P_8. Moreover, the situation concerning $2P_4$ or P_8 is still far from being determined when two forbidden induced subgraphs are considered; in particular, a polynomial-time algorithm for 3-Coloring is not known for (P_8, C_3)-free, (P_8, C_5)-free, $(2P_4, C_3)$-free, or it was not known for $(2P_4, C_5)$-free graphs[1]. In our work, we resolved the last mentioned problem, which considers $(2P_4, C_5)$-free graphs. As mentioned in Chapter 1, a $(2P_4, C_5)$-free graph is clearly not 3-colorable or one of the following is true: the graph is either perfect (and hence we use the existing algorithm), or contains an induced cycle on seven or nine vertices, which enables us to split the analysis further. We analyze the second case,

[1]First two cases were explicitly mentioned as open in [104] and [10], the latter two cases are open to the best of our knowledge.

i.e., $(2P_4, C_5, C_7, \overline{C_7}, K_4)$-free graphs, and thus graphs that surely contain an induced C_9, in Section 4.2.1. We analyze the remaining case in Subsection 4.2.3.

4.2 Proof of Theorem 5

We restate the theorem here.

Theorem 5. *3-Coloring is polynomial-time solvable on $(2P_4, C_5)$-free graphs.*

We present an algorithm that begins by checking that the graph is $\overline{C_7}$-free, and that the neighborhood of each vertex induces a bipartite graph, rejecting the instance if the check fails. Note that this check ensures, in particular, that G is K_4-free.

The algorithm then partitions the graph into connected components, solving the 3-Coloring problem for each component separately. From now on, we assume that the graph $G = (V, E)$ is connected, $\overline{C_7}$-free, and each of its vertices has a bipartite neighborhood.

The basic idea of the algorithm is to choose an initial subgraph N_0 of bounded size, try all possible proper 3-colorings of N_0, and analyze how the precoloring of N_0 affects the possible colorings of the remaining vertices.

We let N_1 denote the vertices in $V \setminus N_0$ which are adjacent to at least one vertex of N_0, and we let N_2 be the set $V \setminus (N_0 \cup N_1)$. We will use the notation $N_i(x)$ for $N_i \cap N(x)$, where $N(x)$ is set of neighbors of x in G.

Our algorithm will iteratively color the vertices of G. We will assume that throughout the algorithm, each vertex v has a list $P(v) \subseteq \{1, 2, 3\}$ of *available colors*. We call $P(v)$ the *palette of* v. The goal is then to find a proper coloring of G in which each vertex is colored by one of its available colors. The problem of deciding the existence of such coloring is known as the List-3-Coloring problem, and is a generalization of the 3-Coloring problem.

Whenever a vertex x of G is colored by a color c in the course of the algorithm, we immediately remove c from the palette of x's neighbors. Additionally, if the vertex x is not in N_0, it is then deleted. The vertices in N_0 are kept in G even after they are colored. We then update the list-3-coloring instance using the following *basic reductions*:

- If a vertex y has only one color c' left in $P(y)$, we color it by the color c' and remove c' from the palettes of its neighbors. If $y \notin N_0$, we then delete y.
- If $P(y)$ is empty for a vertex y, the instance of list-3-coloring is rejected.
- If, for a vertex $y \notin N_0$, the size of $P(y)$ is greater than the degree of y, we delete y.
- *Diamond consistency rule*: If y and y' are a pair of nonadjacent vertices such that $P(y) \neq P(y')$, and if $N(y) \cap N(y')$ is not an independent set, then any valid 3-coloring of G must assign the same color to y and y'; we therefore replace both $P(y)$ and $P(y')$ with $P(y) \cap P(y')$.
- *Neighborhood domination rule*: If y and y' are a pair of nonadjacent vertices such that $N(y) \subseteq N(y')$ and $P(y') \subseteq P(y)$, and if y is not in N_0, we delete y.

- If G has a connected component in which every vertex has at most two available colors, we determine whether the component is colorable by reducing the problem to an instance of 2-SAT. If the component can be colored, we remove it from G and continue. Otherwise, we reject the whole instance.

- If a connected component of G is P_4-free, we solve the list-3-coloring problem for this component by Theorem 24. If the component is colorable, we remove it. Otherwise, we reject the whole instance G.

It is clear that the rules are correct in the sense that the instance of list-3-coloring produced by a basic reduction is list-3-colorable if and only if the original instance was list-3-colorable. It is also clear that we may determine in polynomial time whether an instance of list-3-coloring (with fixed N_0) permits an application of a basic reduction, and perform the basic reduction, if available. Throughout the algorithm, we apply the basic reductions greedily as long as possible until we reach a situation where none of them is applicable.

The basic reductions by themselves are not sufficient to solve the 3-coloring problem for G. Our algorithm will sometimes also need to perform branching, i.e., explore several alternative ways to color a vertex or a set of vertices. Formally, this means that the algorithm reduces a given instance G of list-3-coloring to an equivalent set of instances $\{G_1, \ldots, G_k\}$; here saying that a list-3-coloring instance G is *equivalent* to a set $\{G_1, \ldots, G_k\}$ of instances means that G has a solution if and only if at least one of $G_1, \ldots, G_k$ has a solution.

At the beginning of the algorithm, we attach to each vertex v of G the list $P(v) = \{1, 2, 3\}$ of available colors, thereby formally transforming it to an instance of list-3-coloring. The algorithm will then try all possible proper 3-colorings of N_0, and for each such coloring, apply basic reductions as long as any basic reduction is applicable. If this fails to color all the vertices, more complicated reduction steps and further branching will be performed, to be described later.

Overall, the algorithm will ensure that the initial instance G is eventually reduced to a set of at most polynomially many smaller instances, each of which can be transformed to an equivalent instance of 2-SAT, which then can be solved efficiently.

We state here a theorem that handles the list-3-coloring problem on cographs. We will make use of it later in the proof.

Theorem 24 ([104])**.** *The* LIST-3-COLORING *problem on P_4-free graphs can be solved in polynomial time.*

4.2.1 The C_7-free Case

Our choice of N_0 will depend on the structure of G. More precisely, if G contains an induced copy of C_7, we will choose one such copy as N_0. This is by far the most challenging case, and we return to it later.

The case when G is C_7-free can be handled in a simple way, as we now show.

Proposition 25. *The 3-coloring problem for a $(2P_4, C_5, C_7)$-free graph G can be solved in polynomial time.*

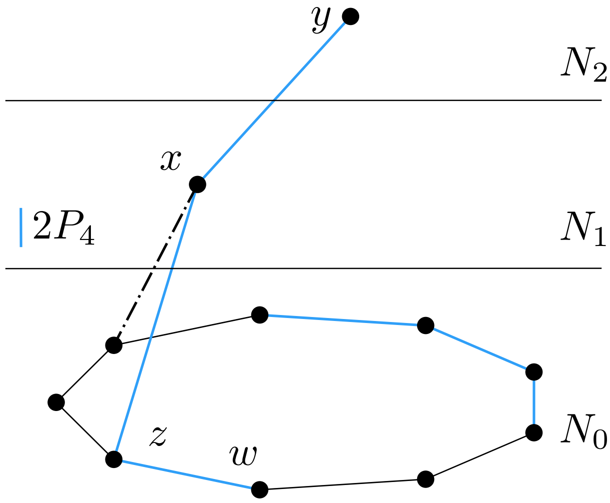


Figure 4.1 Picture showing the induced $2P_4$ in the case of G being C_7-free. If the dash-and-dotted edge is present, A has length 7, otherwise A has length 9.

Proof. Recall that we assume that G is K_4-free and $\overline{C_7}$-free; otherwise G would clearly not be 3-colorable. Note that K_4-freeness implies that G is $\overline{C_k}$-free for every $k \geq 8$, and $2P_4$-freeness implies that G is C_k-free for every $k \geq 10$.

If G is also C_9-free, then it is perfect by the strong perfect graph theorem, and since it is K_4-free, it is 3-colorable. Assume then that G contains an induced copy of C_9. Fix N_0 to be an induced copy of C_9 in G, and define N_1 and N_2 accordingly. We will show that for any proper coloring of N_0, the basic reductions can solve the resulting list-3-coloring problem.

Fix a 3-coloring of N_0, and apply the basic reductions until none of them is applicable. We claim that this solves the instance completely, i.e., we either color the whole graph, or determine that no coloring exists. For contradiction, suppose that we reached a situation when G still contains uncolored vertices, but no basic reduction is applicable.

It follows that G contains a vertex v with three available colors, and this vertex necessarily belongs to N_2. As N_2 is also P_4-free the connected component containing v is not an isolated component within N_2. Therefore, we may find in G two adjacent vertices x, y with $x \in N_1$ and $y \in N_2$. Recall that $N_0(x)$ is the set of vertices of N_0 adjacent to x. The vertices of $N_0(x)$ partition the cycle N_0 into edge-disjoint arcs, and at least one of these arcs has an odd number of edges. Let A be such an arc of odd length.

If A has length 1, then x is adjacent to two adjacent vertices of N_0, hence the color of x is uniquely determined by the coloring of N_0 and x should have been deleted. If A has length 3 or 5, then $A \cup \{x\}$ induces a copy of C_5 or C_7, respectively, which is impossible. Thus, A has length 7 or 9. In such case, we find a copy of $2P_4$ in G, where one P_4 consists of y, x, a vertex $z \in N_0(x)$, and a vertex $w \in A$ adjacent to z, while the other P_4 is formed by taking four consecutive internal vertices of A, each of which is at distance at least two from z and w; see Figure 4.1. In all cases we get a contradiction. □

From now on, we assume that the graph G contains an induced C_7. We choose one such C_7 as N_0, and define N_1 and N_2 accordingly.

4.2.2 More Complicated Reductions

We apply the basic reductions described previously whenever an opportunity arises. Now, we outline more complicated reductions which will be applied only in specific situations.

Cut reduction. Suppose $G = (V, E)$ is a connected instance of list-3-coloring. Let $X \subseteq V$ be a vertex cut of G, let C be a union of one or more connected components of $G - X$, and let C_X be the subgraph of G induced by $C \cup X$. Suppose further that the following conditions hold.

- C has at least two vertices.
- X is an independent set in G.
- All the vertices in X have the same palette, which has size 2.
- For any two vertices x, x' in X, we have $N(x) \cap C = N(x') \cap C$.
- The graph C_X is P_4-free.

Assume without loss of generality that all the vertices of X have the palette equal to $\{1, 2\}$. Let us say that a coloring $c\colon X \to \{1, 2\}$ of X is *feasible for* C, if it can be extended into a proper 3-coloring of the list-3-coloring instance C_X. Note that the feasibility of a given coloring can be determined in polynomial time by Theorem 24, because C_X is a cograph.

We distinguish three types of possible colorings of X: the *all-1* coloring colors all the vertices of X by the color 1, the *all-2* coloring colors all the vertices of X by color 2, and a *mixed* coloring is a coloring that uses both available colors on X. Observe that if X admits at least one mixed coloring feasible for C, then every (not necessarily mixed) coloring of X by colors 1 and 2 is feasible for C. This is because when we extend a mixed coloring of X to a coloring of C_X, all the vertices $y \in C$ must receive the color 3. If such a coloring of C exists, we can combine it with any coloring of X by colors 1 and 2.

The *cut reduction* of X and C is an operation that reduces G to a smaller, equivalent list-3-coloring instance, determined as follows. We choose an arbitrary mixed coloring c of X, and check whether it is feasible for C. If it is feasible, we reduce the instance G to $G - C$, leaving the palettes of the remaining vertices unchanged. The new instance is equivalent to the original one since any proper list-3-coloring of $G - C$ can be extended to a coloring of G because all the colorings of X are feasible for C.

If the mixed coloring c is not feasible for C, we know that no mixed coloring is feasible. We then test the all-1 and the all-2 coloring for feasibility. If both are feasible, we reduce the instance G by replacing C with a single new vertex v, with palette $P(v) = \{1, 2\}$, and connecting v to all the vertices of X. Note that the reduced instance is an induced subgraph of the original one. It is easy to see that the reduced instance is equivalent to the original one.

If only one coloring of X is feasible for C, we delete C, color the vertices of X using the unique feasible coloring, and delete the corresponding color from the palettes of the neighbors of X in $G - C$. If no coloring of X is feasible for C, we declare that G is not list-3-colorable.

Neighborhood collapse. Let G be an instance of list-3-coloring, and let v be a vertex of G. Suppose that $N(v)$ induces in G a connected bipartite graph with nonempty partite classes X and Y. Suppose furthermore that all the vertices of X have the same palette P_X, and all the vertices in Y have the same palette P_Y. The *neighborhood collapse* of the vertex v is the operation that replaces X and Y by a pair of new vertices x and y, adjacent to each other and to v, with the property that any vertex of $G - Y$ adjacent to at least one vertex in X will be made adjacent to x, and similarly every vertex adjacent to Y in $G - X$ will be adjacent to y. We then set $P(x) = P_X$ and $P(y) = P_Y$. Informally speaking, we have collapsed the vertices in X to a single vertex x, and similarly for Y and y.

It is clear that the collapsed instance is equivalent to the original one. However, since the new instance is not necessarily an induced subgraph of the original one, it might happen, e.g., that a collapse performed in a C_5-free graph will introduce a copy of C_5 in the collapsed instance. In our algorithm, we will only perform collapses at a stage when C_5-freeness is no longer needed.

On the other hand, $2P_4$-freeness is preserved by collapses, as we now show.

Lemma 26. *Let G be a $2P_4$-free instance of list-3-coloring in which a neighborhood collapse of a vertex v may be performed, and let G^* be the graph obtained by the collapse. Then G^* is $2P_4$-free.*

Proof. Suppose G^* contains an induced $2P_4$, and let P and Q be the two nonadjacent copies of P_4. Let x and y be the two vertices obtained by collapsing sets X and Y, as in the definition of neighborhood collapse. Without loss of generality, P contains the vertex x. It follows that Q contains none of x, y or v, and in particular, Q is also a P_4 in G.

If the path P contains the edge xy, we may 'lift' P into the graph G by replacing the vertices x and y by appropriate vertices $x' \in X$ and $y' \in Y$, and by replacing the edge xy by a shortest path from x' to y' in $N(v)$. This transforms P into an induced path P' in G on at least four vertices which is nonadjacent to Q. Thus, G also contains a $2P_4$.

Suppose now that P does not contain the edge xy, and therefore y is not in P. If x is the end-vertex of P, say $P = xw_1w_2w_3$, we easily obtain a $2P_4$ in G by simply replacing x by a vertex $x' \in X$ adjacent to w_1 in G. Suppose then that x is an internal vertex of P, say $P = w_1xw_2w_3$. Since we know that P does not contain y, we may replace the vertex w_1 with v in P, knowing that vxw_2w_3 is also an induced P_4 in G^* nonadjacent to Q. By replacing the vertex x by a vertex $x' \in X$ adjacent to w_2, we obtain the induced path $vx'w_2w_3$ in G which forms a $2P_4$ together with Q. □

4.2.3 Graphs Containing C_7

We now turn to the most complicated part of our coloring algorithm, which solves the 3-coloring problem for a $(2P_4, C_5)$-free graph G that contains an induced C_7. We let N_0 be an induced copy of C_7 in this graph, and define N_1 and N_2 accordingly.

We let $v_1, v_2, \ldots, v_7$ denote the vertices of N_0, in the order in which they appear on the cycle N_0. We evaluate their indices modulo 7, so that, e.g., $v_8 = v_1$.

Fix a proper coloring of N_0, and apply the basic reductions to G until no basic reduction is applicable. We now analyze the structure of G at this stage of the algorithm.

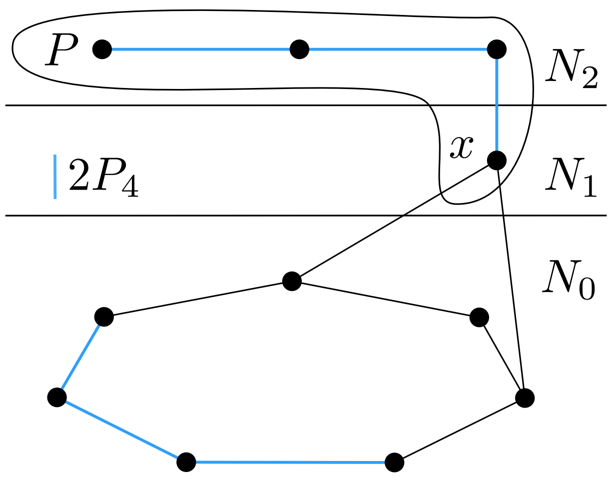


Figure 4.2 Finding an induced $2P_4$, assuming P is an induced P_4 with exactly three vertices in N_2. Note that P can look differently, but always contains x.

We again let $N_0(x)$ denote the set of neighbors of x in N_0.

Lemma 27. *After fixing the coloring of N_0 and applying all available basic reductions, the graph G has the following properties.*

- *Each vertex x of N_1 satisfies either $N_0(x) = \{v_i\}$ for some i, or $N_0(x) = \{v_i, v_{i+2}\}$ for some i.*
- *Each induced copy of P_4 in G has at most two vertices in N_2.*
- *G is connected.*

Proof. To prove the first part, use the vertices of $N_0(x)$ to partition the cycle N_0 into edge-disjoint arcs. Note that none of these arcs has length 1. Since then, x would be adjacent to two vertices of distinct colors, and it would have been colored and deleted. Also, none of these arcs can have length 3, since such an arc together with the vertex x would induce a C_5 in G, contradicting C_5-freeness.

On the other hand, at least one of the arcs formed by $N_0(x)$ must have an odd length. Thus, there is either an arc of length 7, implying $N_0(x) = \{v_i\}$, or there is an arc of length 5, implying $N_0(x) = \{v_i, v_{i+2}\}$ for some i. This proves the first part of the lemma.

To prove the second part, assume that P is an induced copy of P_4 in G with at least three vertices in N_2. If P is fully contained in N_2, then P forms a $2P_4$ together with any P_4 contained in N_0. Suppose that $P \setminus N_2$ consists of a single vertex x, as in Figure 4.2. Necessarily x is in N_1, and by the first part of the lemma, $N_0 \setminus N_0(x)$ contains an induced P_4 which forms an induced $2P_4$ with P.

To prove the last part of the lemma, note that N_0 is connected and therefore contained in a single component of G, and if G contained another connected component, then this other component would necessarily be P_4-free and would be colored by a basic reduction. □

Lemma 27 is the last part of the proof that makes use of the C_5-freeness of G. From now on, we will not need to use the fact that G is C_5-free. In particular, we will allow

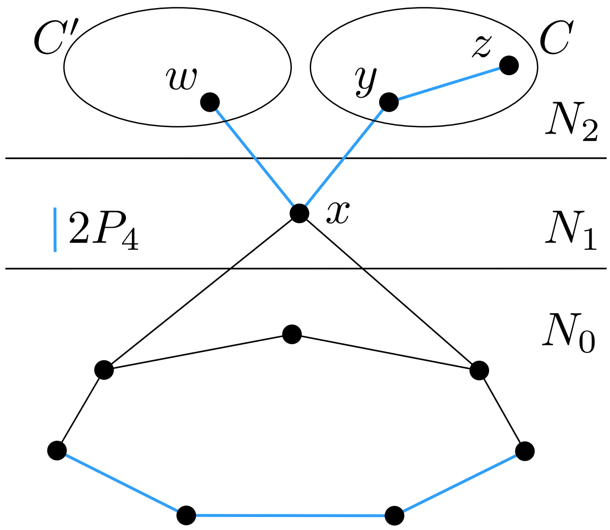


Figure 4.3 Vertex $x \in N_1$ being a partial neighbor of a top component C and neighboring another top component C' leads to an induced $2P_4$.

ourselves reduction operations, such as the neighborhood collapse, which do not preserve C_5-freeness.

We will assume, without mentioning explicitly, that after performing any modification of the list-3-coloring instance G, we always apply basic reductions until no more basic reductions are available.

In the rest of the proof, we use the term *top component* to refer to a connected component of N_2. Observe that every top component is P_4-free and therefore has a dominating set of size at most 2 [63]. We say that a top component is *relevant*, if it contains a vertex z with $|P(z)| = 3$. Note that if G has no relevant top component, then all its vertices have at most two available colors, and the coloring problem can be solved by a single basic reduction.

We will say that a vertex x of N_1 is *relevant* if x is adjacent to a vertex belonging to a relevant top component.

Let $x \in N_1$ be a vertex, and let C be a top component. We say that x is a *partial neighbor* of C, if x is adjacent to at least one but not all the vertices of C. We say that x is a *full neighbor* of C, if it is adjacent to every vertex of C.

Lemma 28. *Suppose $x \in N_1$ is a partial neighbor of a top component C. Then x is not a neighbor of any other top component. Moreover, $|N_0(x)| = 2$.*

Proof. Let y and z be two adjacent vertices belonging to C, such that x is adjacent to y but not to z. Suppose for contradiction that there is a vertex $w \in N_2$ adjacent to x but not belonging to C. Then $wxyz$ is a copy of P_4 with three vertices in N_2, as shown in Figure 4.3, which contradicts Lemma 27. This shows that x is not adjacent to any top component other than C.

Suppose now that $N_0(x)$ contains a single vertex v_i. Then v_ixyz together with $v_{i+2}v_{i+3}v_{i+4}v_{i+5}$ induce a $2P_4$. ☐

We will now reduce G to a set of polynomially many instances in which the set of relevant vertices has a special form. We first eliminate the relevant vertices that have only

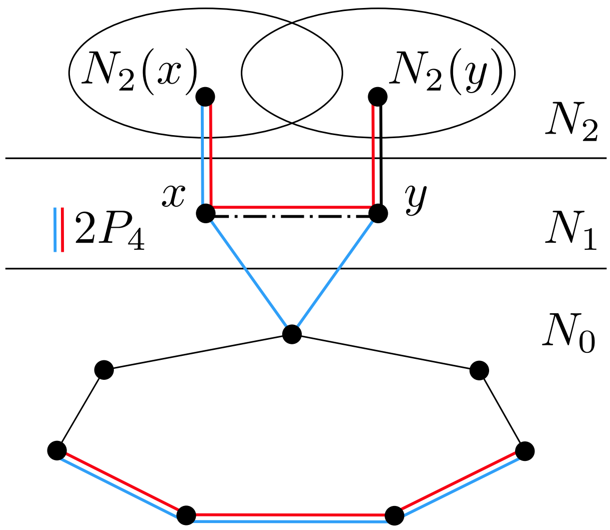


Figure 4.4 The situation obtained from the assumption that $N_2(x)$ and $N_2(y)$ for $x, y \in R_i$ are not comparable by inclusion. The dash-and-dot line represents an edge which is present in one case (red induced $2P_4$) and absent in the other (blue induced $2P_4$).

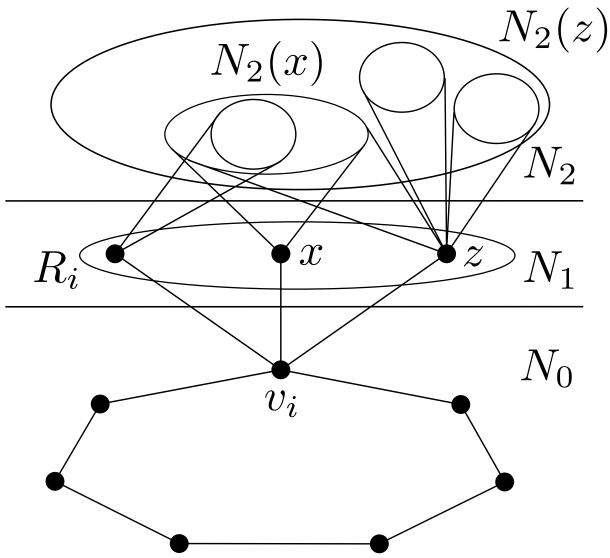


Figure 4.5 The situation obtained when the neighborhoods of vertices in R_i in N_2 are comparable by inclusion with $N_2(z)$ being the largest neighborhood. Note that these vertices are full neighbors of their top components.

one neighbor in N_0. Let R_i be the set of relevant vertices that are adjacent to v_i and not adjacent to any other vertex of N_0.

Lemma 29. *For any $i \in \{1, \ldots, 7\}$, we can reduce G to an equivalent set of at most two instances, both of which satisfy $R_i = \emptyset$.*

Proof. By Lemma 28, we know that any vertex $x \in R_i$ is a full neighbor of each of its adjacent top components.

Let x, y be two distinct vertices of R_i. We claim that the two sets $N_2(x)$ and $N_2(y)$ are comparable by inclusion. To see this, suppose for contradiction that there are vertices

$x' \in N_2(x) \setminus N_2(y)$ and $y' \in N_2(y) \setminus N_2(x)$. Then we can find in G a copy of $2P_4$ in which the first P_4 is $v_{i+2}v_{i+3}v_{i+4}v_{i+5}$, and the second P_4 is either $x'xyy'$ (if $xy \in E(G)$), or $x'xv_iy$ (if $xy \notin E(G)$); see Figure 4.4.

Choose $z \in R_i$ so that $N_2(z)$ is as large as possible. In particular, for every $x \in R_i$, we have $N_2(x) \subseteq N_2(z)$. We then obtain two instances equivalent to G by coloring z by its two available colors. Note that by coloring z, we ensure that all the vertices $N_2(z)$ have at most two available colors, and since z is a full neighbor of all its adjacent top components, this ensures that the vertices of R_i will no longer be relevant after z has been colored; see Figure 4.5 for illustration. □

From now on, we deal with instances of G where every relevant vertex has exactly two neighbors in N_0. Let S_i be the set of relevant vertices adjacent to v_i.

Lemma 30. *For any $i \in \{1, \ldots, 7\}$, we can reduce G to an equivalent set of polynomially many instances, each of which satisfies $S_i = \emptyset$ or $S_{i+3} = \emptyset$.*

Proof. Suppose that the vertices in S_i have available colors 1 and 2, while the vertices in S_{i+3} have available colors 2 and 3 (the case when the vertices in S_{i+3} have the same available colors as the vertices in S_i is similar and we omit it).

For a pair of vertices $x \in S_i$ and $y \in S_{i+3}$, we distinguish the following three possibilities:

(α) $N_2(x)$ and $N_2(y)$ are comparable by inclusion,

(β) x is adjacent to y, or

(γ) neither of the previous two conditions holds.

We say that the pair (x, y) is of *type* α if it satisfies the condition (α) above, and similarly for the other two types. Observe that if the pair (x, y) is of type γ, then there exist $x' \in N_2(x) \setminus N_2(y)$ and $y' \in N_2(y) \setminus N_2(x)$. Moreover, for any choice of such x' and y', the pair $x'y'$ must be an edge of G, otherwise $x'xv_iv_{i-1}$ and $y'yv_{i+3}v_{i+4}$ would form a copy of $2P_4$. In particular, x' and y' belong to the same top component C, and both x and y are partial neighbors of C, as is depicted in Figure 4.6.

Let $Z = S_i \cup S_{i+3}$. Let m be the size of Z, and let us order the vertices of Z into a sequence $z_1, z_2, \ldots, z_m$ satisfying $|N_2(z_1)| \geq |N_2(z_2)| \geq \cdots \geq |N_2(z_m)|$.

We will reduce G to the set of all the instances that can be constructed by one of the following two rules:

(a) All the vertices in Z are colored by their available color different from 2 (i.e., the vertices of S_i are colored by 1, the vertices of S_{i+3} by 3).

(b) Fix a $j \in \{1, \ldots, m\}$ and proceed as follows: color the vertices $z_1, \ldots, z_{j-1}$ by their available color different from 2, and color z_j by 2. Moreover, if z_j is a partial neighbor of a top component C (note that by Lemma 28, z_j is not adjacent to any other top component), color a dominating set of size two in C, in all the possible ways.

We now verify that in all the colorings described above, after all possible basic reductions are applied, either S_i or S_{i+3} becomes empty. This is clearly the case for the coloring

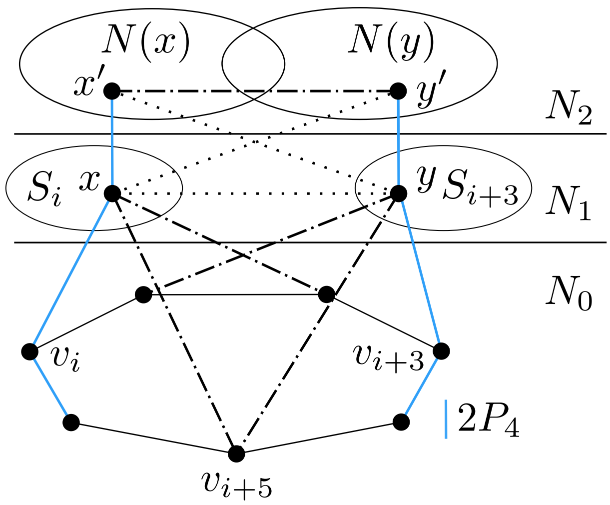


Figure 4.6 Considering a pair of vertices (x, y) of type γ for $x \in S_i, y \in S_{i+3}$, the edge $x'y'$ must be present, otherwise we obtain an induced $2P_4$. The other dash-and-dotted edges are not necessarily present, and the vertex v_{i+5} is adjacent to at most one vertex from $\{x, y\}$.

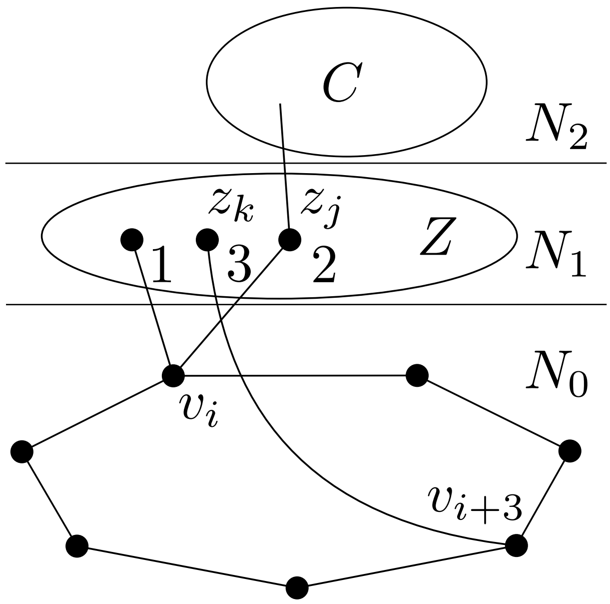


Figure 4.7 Coloring (b) from the proof of Lemma 30 for $k < j$.

described in (a), in which all the vertices in $S_i \cup S_{i+3}$ will be removed from G, so both sets will be empty.

Consider now a coloring described in (b), and assume without loss of generality that z_j is in S_i. We claim that after the coloring is performed, there will be no relevant vertex left in S_{i+3}. To see this, consider a vertex $z_k \in S_{i+3}$. If $k < j$, then z_k has been colored by the color 3, see Figure 4.7.

If $k > j$, we distinguish three possibilities depending on the type of the pair (z_j, z_k). If the pair (z_j, z_k) is of type α, then $N_2(z_k) \subseteq N_2(z_j)$ (recall that $k > j$ implies $|N_2(z_j)| \geq |N_2(z_k)|$). In particular, all the vertices in any top component adjacent to z_k will only have two available colors (recall that if z_j is a partial neighbor of a top component, we

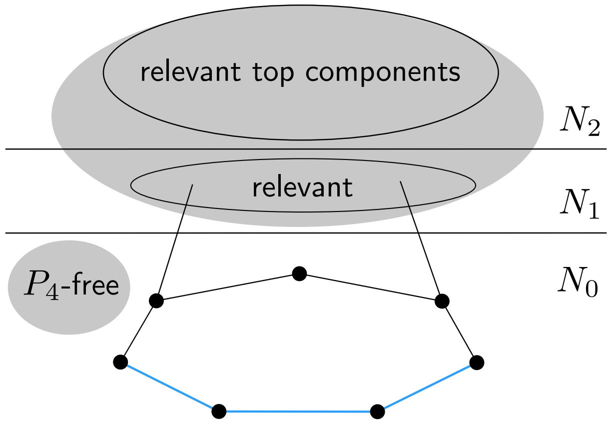


Figure 4.8 The situation considered in the remaining part of the main proof starting with Lemma 31.

also color a dominating set of this top component, ensuring all its vertices have at most two available colors). Thus, z_k will no longer be relevant. If the pair (z_j, z_k) is of type β, i.e. $z_j z_k$ is an edge, then z_k has only the color 3 available and can be colored. Finally, suppose (z_j, z_k) is of type γ. As discussed before, this means both z_j and z_k are partial neighbors of a top component C and have no other neighbors in N_2. After the coloring is performed, all the vertices in C will have only two available colors, because we have colored its dominating set of size two. Hence z_k is no longer relevant. We conclude that S_{i+3} becomes empty, as claimed.

It is clear that the coloring rules (a) and (b) admit only polynomially many possible colorings, and that any valid list coloring of G extends one of the partial colorings described in (a) or in (b). Thus, we reduced G to an equivalent set of polynomially many instances. ☐

From now on, assume that we deal with an instance G in which for every i, one of the two sets S_i and S_{i+3} is empty. Unless the instance is already completely solved, there must be at least one relevant vertex. Assume without loss of generality that G has a relevant vertex adjacent to v_1 and v_3. It follows that S_1 and S_3 are nonempty, and hence S_4, S_5, S_6 and S_7 are empty. Moreover, as any relevant vertex is adjacent to a pair of vertices of the form $\{v_i, v_{i+2}\}$, it follows that S_2 is empty as well. In particular, every relevant vertex x satisfies $N_0(x) = \{v_1, v_3\}$. It follows that all the relevant vertices have the same palette of size 2; assume without loss of generality that this palette is $\{1, 2\}$.

We will now focus on describing the structure of the subgraph of G induced by the relevant vertices and the relevant top components adjacent to them. Let R denote the set of relevant vertices. Note that the subgraph of G induced by $R \cup N_2$ does not contain P_4, otherwise we could use the path $v_4v_5v_6v_7$ to get a $2P_4$ in G; see Figure 4.8.

Note also that if two relevant vertices x and y are adjacent, then any common neighbor of x and y must be colored by color 3, thanks to the diamond consistency rule. We thus know that adjacent relevant vertices have no common neighbors outside N_0. We may also assume that the graph induced by the relevant vertices is bipartite, which we checked as a second step at the beginning of our algorithm. Otherwise, we rejected such an instance as G was clearly not 3-colorable.

Lemma 31. *Suppose that x and y are two adjacent relevant vertices. Let us write $X' = N_2(x)$ and $Y' = N_2(y)$. Then there are disjoint sets $X, Y \subseteq R$, with $x \in X$ and $y \in Y$, satisfying these properties:*

1. *Every vertex in $X \cup Y'$ is adjacent to every vertex in $Y \cup X'$.*
2. *X and Y are independent sets of G.*
3. *The vertices in $X' \cup Y'$ are only adjacent to vertices in $X \cup Y \cup X' \cup Y'$; in particular, $X' \cup Y'$ induce a top component.*

Proof. Consider the subgraph $G[R]$ of G induced by the relevant vertices, and let C be the connected component of $G[R]$ containing x and y. Recall that C must be bipartite. We let X and Y be its partite classes containing x and y, respectively. Note that C is complete bipartite. Otherwise, it would contain a P_4.

We will now show that all the vertices in X have the same neighbors in N_2. Indeed, if we could find a pair of vertices $x_1, x_2 \in X$ and a vertex $x' \in N_2(x_1)$ not adjacent to x_2, then $x'x_1yx_2$ would induce a P_4. It follows that for every $x_1 \in X$ we have $N_2(x_1) = X'$, and similarly for every $y_1 \in Y$ we have $N_2(y_1) = Y'$.

We saw that adjacent relevant vertices have no common neighbors in N_2, so X' and Y' are disjoint. Every vertex in X' must be adjacent to every vertex in Y', for if there were nonadjacent vertices $x' \in X'$ and $y' \in Y'$, then $x'xyy'$ would induce a P_4. This proves the first claim of the lemma.

To prove the second claim, observe that X and Y are independent by construction.

To prove the third claim, proceed by contradiction and assume that a vertex $x' \in X' \cup Y'$ is adjacent to a vertex z not belonging to $X \cup Y \cup X' \cup Y'$. We may assume that x' belongs to X'. Necessarily, z belongs to $R \cup N_2$, and $zx'xy$ induces a forbidden P_4. □

Suppose $G[R]$ contains at least one edge xy, and let X, Y, X', Y' be as in the previous lemma. Note that there are only two possible ways to color $G[X \cup Y]$ – either X is colored 1 and Y is colored 2, or vice versa. We can check in polynomial time which of these two colorings can be extended to a valid coloring of $G[X \cup Y \cup X' \cup Y']$. If neither of the two colorings extends, we reject the list 3-coloring instance, if only one of the two colorings extends, we color $X \cup Y$ accordingly, and if both colorings extend, we remove the vertices $X' \cup Y'$ from G, resulting in a smaller equivalent instance, in which $X \cup Y$ is no longer relevant. Repeating this for every component of $G[R]$ that contains at least one edge, we eventually reduce the problem to an instance in which the relevant vertices form an independent set.

From now on, we assume R is independent in G. For a vertex $x \in R$, let $C_2(x)$ denote the set of top components that contain at least one neighbor of x.

Lemma 32. *For any two relevant vertices x and y, we either have $C_2(x) = C_2(y)$, or $C_2(x)$ and $C_2(y)$ are disjoint.*

Proof. Suppose the lemma fails for some x and y. We may then assume that there is a top component $C \in C_2(x) \cap C_2(y)$ and a component $C' \in C_2(x) \setminus C_2(y)$. Since $|C_2(x)| \geq 2$, we know from Lemma 28 that x is a full neighbor of all the top components in $C_2(x)$. Choose a vertex $u \in C'$ and a vertex $v \in C \cap N_2(y)$. Then $uxvy$ is a copy of P_4 in $R \cup N_2$, which is impossible. □

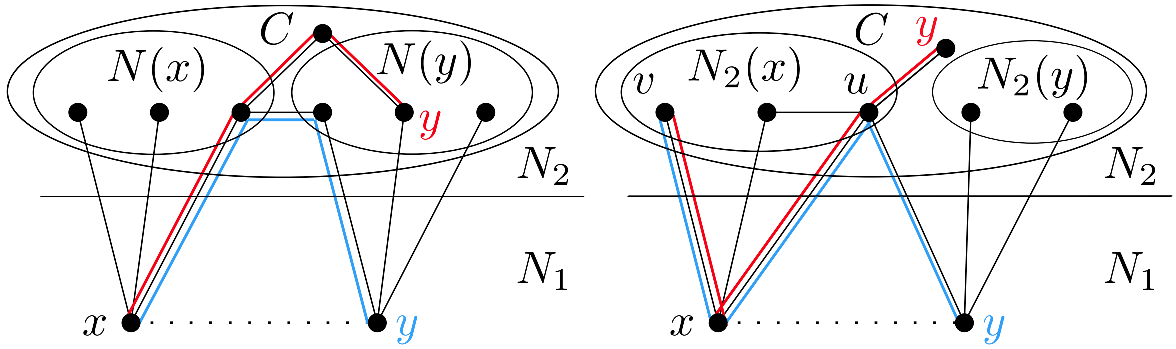


Figure 4.9 Illustrations to the proof of Lemma 33. The left part shows a situation when y is not adjacent to any vertex in $N_2(x)$, the right part shows a situation when y has a neighbor in $N_2(x)$ which is disconnected. Each part depicts two different possibilities. The blue P_4 shows the case $y \in R_x$, while the red P_4 shows the case when $y \in C$.

Let us say that two relevant vertices x and y are *equivalent* if $C_2(x) = C_2(y)$. As the next step in our algorithm, we will process the equivalence classes one by one, with the aim to reduce the instance G to an equivalent instance in which each relevant vertex is adjacent to a single top component.

Let $x \in R$ be a vertex such that $|C_2(x)| \geq 2$, and let R_x be the equivalence class containing x. By Lemma 28, each vertex in R_x is a full neighbor of any component in $C_2(x)$, and by Lemma 32, no vertex outside of R_x may be adjacent to a relevant top component in $C_2(x)$. Thus, R_x is a vertex cut separating the relevant top components in $C_2(x)$ from the rest of G. We may therefore apply the cut reduction through the vertex cut R_x to reduce G to a smaller instance in which the vertices of R_x are no longer relevant.

We repeat the cut reductions until there is no relevant vertex adjacent to more than one top component. From now on, we deal with instances in which each relevant vertex is adjacent to a unique top component; note that this top component is necessarily relevant.

Lemma 33. *Let x be a relevant vertex, let C be the top component adjacent to x, let R_x be the equivalence class of x, and let $y \in R_x \cup C$ be a vertex not adjacent to x. Then y is adjacent to at least one vertex in $N_2(x)$. Moreover, if $N_2(x)$ induces a disconnected subgraph of G, then y is adjacent to all the vertices of $N_2(x)$.*

Proof. Refer to Figure 4.9. If y is not adjacent to any vertex of $N_2(x)$, then we can find an induced path with at least four vertices by considering the shortest path from x to y in the graph induced by $C \cup \{x, y\}$. Therefore y has at least one neighbor in $N_2(x)$. Suppose now that $N_2(x)$ is disconnected. If y is not adjacent to all the vertices of $N_2(x)$, then we can find a vertex $u \in N_2(x)$ adjacent to y, and a vertex $v \in N_2(x)$ nonadjacent to y, in such a way that u and v are in distinct components of $N_2(x)$. Then $yuxv$ is an induced P_4. □

Fix now a relevant top component C and let R be set of relevant vertices in N_1 adjacent to C. Fix a vertex $x \in R$ so that $N_2(x)$ is as large as possible. Let R_x be the equivalence class containing x. We distinguish several possibilities, based on the structure of $N_2(x)$.

$N_2(x)$ is disconnected. Suppose first that $N_2(x)$ induces in G a disconnected subgraph. By Lemma 33, any vertex in R_x is adjacent to all vertices in $N_2(x)$. By our choice of x,

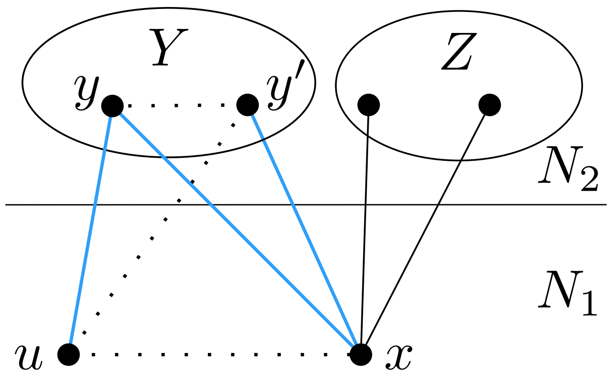


Figure 4.10 There is an induced P_4 in N_2 if $N_2(x)$ is connected with ≥ 3 vertices and for $y, y' \in Y$ there exists a u neighboring only one of them.

this implies that for any $x' \in R_x$ we have $N_2(x') = N_2(x)$. We may therefore apply the cut reduction for the cut R_x that separates C from the rest of G, to obtain a smaller instance in which the vertices of R_x are no longer relevant.

$N_2(x)$ **is connected, with** ≥ 3 **vertices.** Now suppose that $N_2(x)$ induces a connected graph, and that $N_2(x)$ has at least three vertices. We now verify that $N_2(x)$ induces a complete bipartite graph, otherwise C contains P_4 or G is not 3-colorable. Let Y and Z be the two partite classes of $N_2(x)$. Note that any two vertices y, y' in Y have the same neighbors in G: indeed if u were a vertex adjacent to y but not to y', then $uyxy'$ would induce a copy of P_4, as depicted in Figure 4.10. By the same argument, all the vertices in Z have the same neighbors in G as well. Diamond consistency enforces that all the vertices in Y have the same palette, and similarly for Z. We may then invoke neighborhood domination to delete from Y all vertices except a single vertex y, and do the same with Z, reducing G to an equivalent instance in which $N_2(x)$ consists of a single edge.

$N_2(x)$ **is a single vertex.** Suppose that $N_2(x)$ consists of a single vertex y. If y is the only vertex of C, then y must have the palette $\{1, 2, 3\}$, otherwise C would not be a relevant component. In such case, we may simply color y with color 3 and delete it, as this does not restrict the possible colorings of $G - y$ in any way. If, on the other hand, C has more than one vertex, it follows from Lemma 33 that all the vertices of R_x are adjacent to y, and by the choice of x, every vertex in R_x is adjacent to y as its only neighbor in C. We may then apply cut reduction for the cut R_x. In all cases, we obtain a smaller equivalent instance, in which the vertices in R_x are no longer relevant.

$N_2(x)$ **is a single edge.** The last case to consider deals with the situation when $N_2(x)$ contains exactly two adjacent vertices u and v. Assume that $\deg_G(u) \geq \deg_G(v)$. Recall that the set R of relevant vertices is independent. Note that for any vertex $x' \in R_x$, $N_2(x')$ is connected, otherwise Lemma 33 implies that $N_2(x')$ is contained in $N_2(x)$, contradicting $N_2(x)$ being a single edge.

We first claim that any vertex $y \in R_x \cup (C - u)$ adjacent to v is also adjacent to u. Suppose this is not the case. Then, since $\deg_G(u) \geq \deg_G(v)$, there must also be a vertex

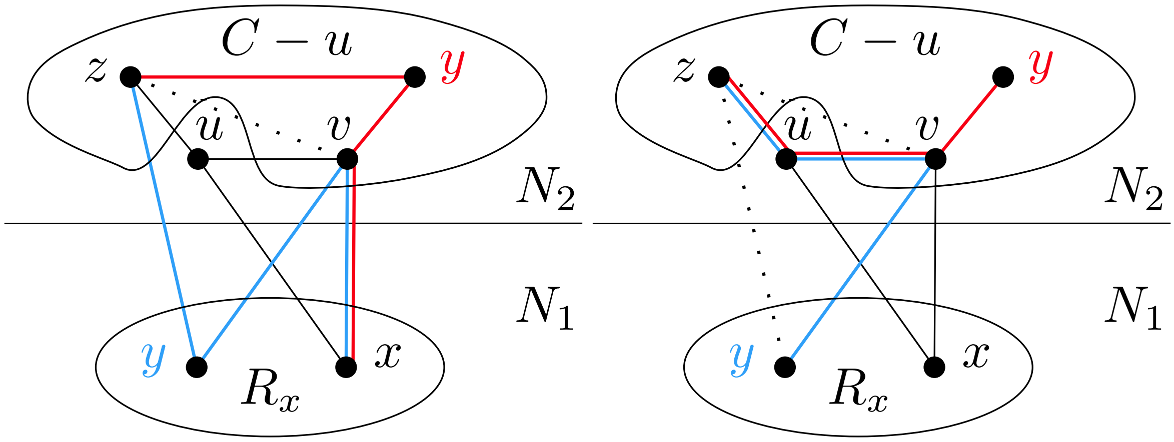


Figure 4.11 Illustrations of the situation when $N_2(x)$ is a single edge uv. Again, case $y \in R_x$ is shown as blue y and blue P_4, while $y \in (C - u)$ is shown as red y and red P_4. There are two subcases corresponding to yz being an edge or not.

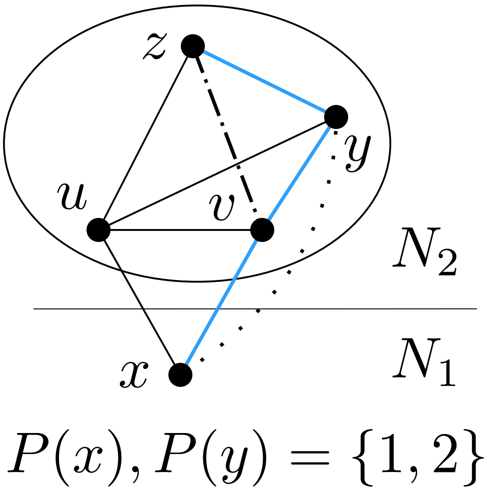


Figure 4.12 Recall that u is adjacent to all vertices of C. If there is a $y \in C$ adjacent to both u and v, there is no other neighbor z of y. Otherwise, either zv is an edge, causing a K_4, or zv is not an edge, causing an induced P_4.

$z \in R_x \cup (C - v)$ adjacent to u but not to v. If yz is an edge, then $zyvx$ is a copy of P_4, and if yz is not an edge, then $zuvy$ is a copy of P_4, as shown in Figure 4.11. In both cases we have a contradiction, establishing the claim. Note that the claim, together with Lemma 33, implies that u is adjacent to all the other vertices of $R_x \cup C$.

Next, we show that if C contains a vertex adjacent to both u and v, then we may reduce G to a smaller equivalent instance. Suppose $y \in C$ is adjacent to u and v. Then $P(y) = P(x) = \{1, 2\}$ by diamond consistency. We now claim that y has no other neighbors in G beyond u and v. Suppose that $z \notin \{u, v\}$ is a neighbor of y. Then z cannot be adjacent to v, since $uvyz$ would form a K_4. Therefore $zyvx$ is a copy of P_4, a contradiction illustrated by Figure 4.12. We conclude that $N(y) = \{u, v\} \subseteq N(x)$, and since $P(y) = P(x)$, we may delete y due to neighborhood domination.

From now on, we assume that u and v have no common neighbor in C. Recall that u is adjacent to all the other vertices in $C \cup R_x$. We now reduce G to an instance where

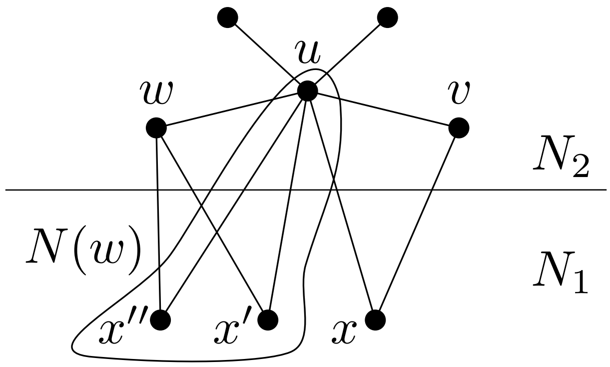


Figure 4.13 Situation in which we can apply a neighborhood collapse.

$C - u$ is an independent set. We already know that v is isolated in $C - u$ by the previous paragraph. Suppose that $C - u$ has a component D with more than one vertex. Suppose D has a vertex v' adjacent to a vertex $x' \in R_x$. Let y' be any vertex in $N(v') \cap D$. Observe that $x'y'$ is not an edge as otherwise $x'y'v'u$ is a K_4. Hence, $x'y'v'u$ form a diamond. Then $P(y') = P(x')$ by diamond consistency. We claim that y' has no other neighbors in G beyond u and v'. Suppose that $z' \notin \{u, v'\}$ is a neighbor of y'. Then z' cannot be adjacent to v', since $uv'y'z'$ would form a K_4. If $z'x'$ is not an edge $z'y'v'x'$ is a copy of P_4, a contradiction. If $z'x'$ is an edge then all $z'v'u$ are in $N_2(x)$ contradicting the choice of x so that $N_2(x)$ is the largest possible.

we can repeat the reasoning of the previous paragraph with x' and v' in the place of x and v, showing that u and v' cannot have any common neighbor in C, contradicting the assumption that D has more than one vertex. We can thus conclude that D is not adjacent to any vertex in R_x. Then u is a cut-vertex separating D from the rest of G. We may test which colorings of u can be extended into D (since D is P_4-free, this can be done efficiently), then restrict the palette of u to only the feasible colors, and then delete D.

We are now left with a situation when C is a star with center u, and every vertex of R_x is adjacent to u and to at most one vertex of $C - u$. If there is a vertex $w \in C - u$ adjacent to more than one vertex in R_x, it means that the neighborhood of w is a connected bipartite graph (a star with center u) to which we may apply neighborhood collapse; see Figure 4.13.

Suppose now that every vertex $w \in C - u$ has only one neighbor in R_x (if w had no neighbor in R_x, it would have degree 1 and we could remove it). If w's palette has 3 colors, we can remove it, so we may assume that every vertex in $C - u$ has a palette of size 2. Then u's palette must have 3 colors. Otherwise, C would not be a relevant component. If a vertex in $C - u$ has palette $\{1, 2\}$, then u must be colored 3 and then R_x is no longer relevant.

It remains to consider the case when each vertex of $C - u$ has the palette $\{1, 3\}$ or $\{2, 3\}$. Let W_1 and W_2 be the sets of vertices of $C - u$ having palette $\{1, 3\}$ and $\{2, 3\}$, respectively. Let X_1 and X_2 be the sets of vertices of R_x that are adjacent to a vertex in W_1 and W_2, respectively. Let X_0 be the set of vertices in R_x that have no neighbor in $C - u$. The situation is shown in Figure 4.14. Let us consider the possible colorings of $C \cup R_x$. If u is colored by 3, then the whole set W_1 is colored by 1, W_2 is colored by 2, hence X_1 is colored by 2 and X_2 by 1, while the vertices in X_0 can be colored arbitrarily by 1 or 2. On the other hand, if u receives a color $\alpha \neq 3$, then all the vertices in R_x

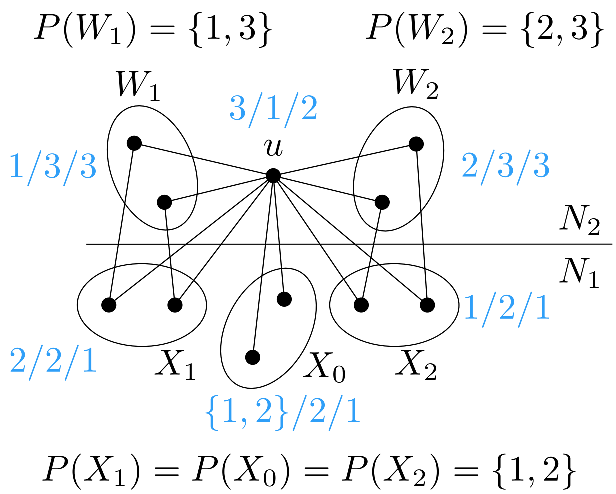


Figure 4.14 The last case in which each vertex of $C - u$ has palette $\{1, 3\}$ or $\{2, 3\}$. The blue text represents the three possibilities to color u and what colors that implies for other parts of the graph.

receive the color $\beta \in \{1, 2\} \setminus \{\alpha\}$, and the vertices in $C - u$ can be colored by 3. The set R_x therefore admits three types of feasible colorings: the all-1 coloring, the all-2 coloring, and any coloring where the set X_1 is colored by 2 and X_2 by 1. This set of colorings can be equivalently characterized by the following properties:

- If a vertex in X_1 is colored by 1, then the whole R_x receives 1.
- If a vertex in X_2 is colored by 2, then the whole R_x is colored by 2.
- All the colors in X_1 are equal and all the colors in X_2 are equal.

The above properties can be encoded by a 2-SAT formula whose variables correspond to vertices of R_x.

To summarize, we have shown that a 3-coloring instance G can be reduced to an equivalent set of polynomially many simpler list-3-coloring instances. The structure of these simpler instances guarantees that for any relevant top component C, we can form a 2-SAT formula describing the colorings of the relevant vertices adjacent to C that can be extended to a proper coloring of C. Moreover, in the subgraph of G induced by the vertices not belonging to any relevant top component, each vertex has a palette of size at most two. The colorings of this subgraph can again be encoded by a 2-SAT formula. Such an instance of list-3-coloring then admits a solution if and only if there is a satisfying assignment for the conjunction of the 2-SAT formulas described above. The existence of such an assignment can be found in polynomial time. This completes the proof of Theorem 5.

4.3 Conclusions

We have shown that 3-Coloring on $(2P_4, C_5)$-free graphs is solvable in polynomial time. As we discussed in the introduction, this approach might serve as a step towards resolving 3-Coloring on $2P_4$-free graphs because it remains to consider $2P_4$-free graphs containing C_5.

Apart from the main question above, under more refined scale, the complexity of 3-Coloring on $(2P_4, C_3)$-free, (P_8, C_3)-free, or (P_8, C_5)-free graphs remains unknown. In another direction, it would be interesting to extend our result to the list 3-coloring setting.

5 Clique-Width: Harnessing the Power of Atoms

5.1 Introduction

We recall the main question that motivates our research.

> Which hereditary graph classes of *unbounded* clique-width have the property that their atoms have *bounded* clique-width?

Several classical graph problems, such as Coloring, Minimum Fill-In, Maximum Clique, Maximum Weighted Independent Set [168, 8] and Maximum Induced Matching [38] are polynomial-time solvable on a hereditary graph class $\mathcal{G}$ if and only if this is the case on the atoms of $\mathcal{G}$.

As mentioned in Chapter 1, if a class defined by one forbidden induced subgraph is considered, the restriction to atoms does not yield any algorithmic advantages, as shown by Gaspers et al. [101].

Theorem 34 ([101])**.** *Let H be a graph. The class of H-free* ***atoms*** *has bounded clique-width if and only if the class of H-free* ***graphs*** *has bounded clique-width (so, if and only if H is an induced subgraph of P_4).*

Therefore, we focus on bigenic graph classes, i.e., graph classes defined by two forbidden induced subgraphs. The survey [71] gives a state-of-the-art theorem on the boundedness and unboundedness of clique-width of bigenic graph classes. Unlike treewidth, for which a complete dichotomy is known [15], this state-of-the-art theorem shows that there are still five open cases (up to an equivalence relation); see also Section 5.4. From the same theorem, we observe that many graph classes are of unbounded clique-width.

Except for the mentioned case of (C_4, P_6)-free graphs [101], we are aware of other hereditary graph classes $\mathcal{G}$ with the property of having unbounded clique-width but bounded clique-width of atoms, but in those cases $|\mathcal{F}_{\mathcal{G}}| > 2$. Chordal graphs, or equivalently, $(C_4, C_5, \ldots)$-free graphs have unbounded clique-width [152], but chordal atoms are complete graphs [80] and have clique-width at most 2. The same holds for any subclass of chordal graphs of unbounded clique-width, such as the class of split graphs [152], or equivalently, $(C_4, C_5, 2P_2)$-free graphs. Moreover, Cameron et al. [40] proved that (cap, C_4)-free odd-signable atoms have clique-width at most 48, whereas the class of all (cap, C_4)-free odd-signable graphs contains the class of split graphs and thus has unbounded clique-width. We refer to [89, 92] for some examples of polynomial-time algorithms for Coloring on hereditary graph classes that exploit the fact that atoms of subclasses of these graph classes have bounded clique-width.

Our Results. Due to Theorem 34, we focus on the atoms of bigenic graph classes. We restate our main theorem about an incomparable bigenic graph class that has atoms of bounded clique-width.

Theorem 6. *The class of $(2P_2, \overline{P_2 + P_3})$-free atoms has bounded clique-width (whereas the class of $(2P_2, \overline{P_2 + P_3})$-free graphs has unbounded clique-width).*

We prove Theorem 6 in Section 5.3 after first giving an outline. Our approach shares some similarities with the approach Malyshev and Lobanova [153] used to show that (Weighted) Coloring is polynomial-time solvable on $(P_5, \overline{P_2 + P_3})$-free graphs. We explain the differences between both approaches and the new ingredients of our proof in detail in Section 5.3.

We also identify a number of new bigenic graph classes whose atoms already have unbounded clique-width. Combining the constructions from Section 5.5 with Theorem 6 and the state-of-art theorem on clique-width from [71] yields the following (restated) summary; see Chapters 2 and 5.2 for the notation used.

Theorem 7. *For graphs H_1 and H_2, let $\mathcal{G}$ be the class of (H_1, H_2)-free graphs.*

1. *The class of atoms in $\mathcal{G}$ has bounded clique-width if*
   - *(i)* H_1 *or* $H_2 \subseteq_i P_4$
   - *(ii)* $H_1 = \text{paw}$ *or* K_s *and* $H_2 = P_1 + P_3$ *or* tP_1 *for some* $s, t \geq 1$
   - *(iii)* $H_1 \subseteq_i \text{paw}$ *and* $H_2 \subseteq_i K_{1,3}+3P_1,\ K_{1,3}+P_2,\ P_1+P_2+P_3,\ P_1+P_5,\ P_1+S_{1,1,2},\ P_2+P_4,\ P_6,\ S_{1,1,3}$ *or* $S_{1,2,2}$
   - *(iv)* $H_1 \subseteq_i P_1 + P_3$ *and* $H_2 \subseteq_i \overline{K_{1,3}+3P_1},\ \overline{K_{1,3}+P_2},\ \overline{P_1+P_2+P_3},\ \overline{P_1+P_5},\ \overline{P_1+S_{1,1,2}},\ \overline{P_2+P_4},\ \overline{P_6},\ \overline{S_{1,1,3}}$ *or* $\overline{S_{1,2,2}}$
   - *(v)* $H_1 \subseteq_i \text{diamond}$ *and* $H_2 \subseteq_i P_1 + 2P_2,\ 3P_1 + P_2$ *or* $P_2 + P_3$
   - *(vi)* $H_1 \subseteq_i 2P_1 + P_2$ *and* $H_2 \subseteq_i \overline{P_1 + 2P_2},\ \overline{3P_1 + P_2}$ *or* $\overline{P_2 + P_3}$
   - *(vii)* $H_1 \subseteq_i \text{gem}$ *and* $H_2 \subseteq_i P_1 + P_4$ *or* P_5
   - *(viii)* $H_1 \subseteq_i P_1 + P_4$ *and* $H_2 \subseteq_i \overline{P_5}$
   - *(ix)* $H_1 \subseteq_i K_3 + P_1$ *and* $H_2 \subseteq_i K_{1,3}$,
   - *(x)* $H_1 \subseteq_i \overline{2P_1 + P_3}$ *and* $H_2 \subseteq_i 2P_1 + P_3$
   - *(xi)* $H_1 \subseteq_i P_6$ *and* $H_2 \subseteq_i C_4$, *or*
   - *(xii)* $H_1 \subseteq_i 2P_2$ *and* $H_2 \subseteq_i \overline{P_2 + P_3}$.

2. *The class of atoms in $\mathcal{G}$ has unbounded clique-width if*
   - *(i)* $H_1 \notin \mathcal{S}$ *and* $H_2 \notin \mathcal{S}$
   - *(ii)* $H_1 \notin \overline{\mathcal{S}}$ *and* $H_2 \notin \overline{\mathcal{S}}$
   - *(iii)* $H_1 \supseteq_i K_3 + P_1$ *and* $H_2 \supseteq_i 4P_1$ *or* $2P_2$
   - *(iv)* $H_1 \supseteq_i K_{1,3}$ *and* $H_2 \supseteq_i K_4$ *or* C_4
   - *(v)* $H_1 \supseteq_i \text{diamond}$ *and* $H_2 \supseteq_i K_{1,3},\ 5P_1,\ P_2 + P_4$ *or* $P_1 + P_6$
   - *(vi)* $H_1 \supseteq_i 2P_1 + P_2$ *and* $H_2 \supseteq_i K_3 + P_1,\ K_5,\ \overline{P_2 + P_4}$ *or* $\overline{P_6}$
   - *(vii)* $H_1 \supseteq_i K_3$ *and* $H_2 \supseteq_i 2P_1 + 2P_2,\ 2P_1 + P_4,\ 4P_1 + P_2,\ 3P_2$ *or* $2P_3$
   - *(viii)* $H_1 \supseteq_i 3P_1$ *and* $H_2 \supseteq_i \overline{2P_1 + 2P_2},\ \overline{2P_1 + P_4},\ \overline{4P_1 + P_2},\ \overline{3P_2}$ *or* $\overline{2P_3}$
   - *(ix)* $H_1 \supseteq_i K_4$ *and* $H_2 \supseteq_i P_1 + P_4,\ 3P_1 + P_2$ *or* $2P_2$
   - *(x)* $H_1 \supseteq_i 4P_1$ *and* $H_2 \supseteq_i \text{gem},\ \overline{3P_1 + P_2}$ *or* C_4

(xi) $H_1 \supseteq_i$ gem, $\overline{P_1 + 2P_2}$ *or* $\overline{P_2 + P_3}$ *and* $H_2 \supseteq_i P_1 + 2P_2$ *or* P_6

(xii) $H_1 \supseteq_i P_1 + P_4$ *and* $H_2 \supseteq_i \overline{P_1 + 2P_2}$*, or*

(xiii) $H_1 \supseteq_i 2P_2$ *and* $H_2 \supseteq_i \overline{P_2 + P_4}$, $\overline{3P_2}$ *or* $\overline{P_5}$.

We prove Theorem 7 in Section 5.6. We remind that due to this theorem, we are left with 18 open cases, which we list in Section 5.6 (see Open Problem 2). In Section 5.7 we discuss directions for future work.

5.2 Preliminaries

We restate the two main definitions of this chapter. A clique $K \subseteq V(G)$ is a *clique cut-set* of G if $G \setminus K$ is disconnected. A graph with no clique cut-sets is an *atom*. Note that if a graph is disconnected, then the empty set is a clique cut-set, so atoms are connected by definition. We let $\mathcal{S}$ denote the class of graphs every connected component of which is either a subdivided claw or a path on at least one vertex.

The following observation is well known and easy to see.

Lemma 35. *A graph is bipartite chain if and only if it is bipartite and* $2P_2$*-free.*

5.2.1 Clique-width and Operations

We refer the reader to Chapter 2 for the definition of clique-width.

For an induced subgraph G' of a graph G, the *subgraph complementation* acting on G with respect to G' replaces every edge of G' by a non-edge, and vice versa. Hence, the resulting graph has vertex set $V(G)$ and edge set $(E(G) \setminus E(G')) \cup E(\overline{G'})$. For two disjoint vertex subsets S and T in G, the *bipartite complementation* acting on G with respect to S and T replaces every edge with one end-vertex in S and the other in T by a non-edge and vice versa.

For a constant $k \geq 0$ and a graph operation γ, a graph class $\mathcal{G}'$ is (k, γ)*-obtained* from a graph class $\mathcal{G}$ if

(i) every graph in $\mathcal{G}'$ is obtained from a graph in $\mathcal{G}$ by performing γ at most k times, and

(ii) for every $G \in \mathcal{G}$, there exists at least one graph in $\mathcal{G}'$ obtained from G by performing γ at most k times.

Then γ *preserves* boundedness of clique-width if for every constant k and every graph class $\mathcal{G}$, every graph class $\mathcal{G}'$ that is (k, γ)-obtained from $\mathcal{G}$ has bounded clique-width if and only if $\mathcal{G}$ has bounded clique-width.

Fact 1. Vertex deletion preserves boundedness of clique-width [147].

Fact 2. Subgraph complementation preserves boundedness of clique-width [130].

Fact 3. Bipartite complementation preserves boundedness of clique-width [130].

We finish this section with making two further observations that we will need later on. First, we make the following well-known observation on bipartite chain graphs, which is readily seen.

Lemma 36. *Bipartite chain graphs have clique-width at most* 3.

Let $G = (K \cup I, E)$ be a split graph with clique K and independent set I. If there is a vertex $v \in I$ with $N(v) \subsetneq K$, then $N(v)$ is a clique cut-set of G. Furthermore, if $|I| > 1$ then K is a clique cut-set. It follows that split atoms are complete graphs. It is readily seen that complete graphs have clique-width at most 2. Hence, we can make the following observation (which also follows from a theorem of Dirac [80] and the fact that every split graph is chordal).

Lemma 37. *Split atoms are complete graphs and have clique-width at most* 2.

5.3 The Proof of Theorem 6

Here, we prove Theorem 6, namely that the class of $(2P_2, \overline{P_2 + P_3})$-free atoms has bounded clique-width. Our approach is based on the following three claims:

(i) $(2P_2, \overline{P_2 + P_3})$-free atoms with an induced C_5 have bounded clique-width.

(ii) $(2P_2, \overline{P_2 + P_3})$-free atoms with an induced C_4 have bounded clique-width.

(iii) $(C_4, C_5, 2P_2, \overline{P_2 + P_3})$-free atoms have bounded clique-width.

We prove Claims (i) and (ii) in Lemmas 38 and 45, respectively, whereas Claim (iii) follows from the fact that $(C_4, C_5, 2P_2)$-free graphs are split graphs and so, by Lemma 37, the atoms in this class are complete graphs and therefore have clique-width at most 2. We partition the vertex set of an arbitrary $(2P_2, \overline{P_2 + P_3})$-free atom G into a number of different subsets according to their neighborhoods in an induced C_5 in Lemma 38 or an induced C_4 in Lemma 45. We then analyse the properties of these different subsets of $V(G)$ and how they are connected to each other, and use this knowledge to apply a number of appropriate vertex deletions, subgraph complementations and bipartite complementations. These operations will modify G into a graph G' that is a disjoint union of a number of smaller "easy" graphs known to have "small" clique-width. We then use Facts 1–3 to conclude that G also has small clique-width.

This approach works, as we will:

- apply the vertex deletions, subgraph complementations, and bipartite complementations only a constant number of times;

- not use the properties of being an atom or being $(2P_2, \overline{P_2 + P_3})$-free once we "leave the graph class" due to applying the above graph operations.

Our approach is similar to the approach used by Malyshev and Lobanova [153] for showing that Coloring is polynomial-time solvable on the superclass of $(P_5, \overline{P_2 + P_3})$-free graphs. However, we note the following two techniques that can be used in the design of algorithms for Coloring on hereditary graph classes, but cannot be used for proving boundedness of clique-width. Both these techniques were used in [153].

1. Prime atoms restriction. A set $X \subseteq V(G)$ is a *module* if all vertices in X have the same set of neighbors in $V(G) \setminus X$. A module X in a graph G is *trivial* if it contains either all or at most one vertex of G. A graph G is *prime* if it has no non-trivial modules. To solve COLORING in polynomial time on some hereditary graph class $\mathcal{G}$, one may restrict to prime atoms in $\mathcal{G}$ [122]. Malyshev and Lobanova proved that $(P_5, \overline{P_2 + P_3})$-free prime atoms with an induced C_5 are $3P_1$-free or have a bounded number of vertices. In both cases, COLORING can be solved in polynomial time. We cannot make the pre-assumption that our atoms are prime. To see this, let G be a split graph that is not complete. Add two new non-adjacent vertices u and v to G and make them complete to the rest of $V(G)$. Let $\mathcal{G}$ be the (hereditary) graph class that consists of all these "enhanced" split graphs and their induced subgraphs. These enhanced split graphs are atoms, which have unbounded clique-width due to Fact 1 and the fact that split graphs have unbounded clique-width [152]. However, the prime atoms in $\mathcal{G}$ are P_1 and P_2,[1] which have clique-width 1 and 2, respectively.

2. Perfect graphs restriction. Malyshev and Lobanova observed that the class of $(P_5, \overline{P_2 + P_3}, C_5)$-free graphs is perfect. Hence, COLORING can be solved in polynomial time on such graphs [107]. However, being perfect does not imply boundedness of clique-width. For instance, split graphs are perfect graphs with unbounded clique-width [152].

Lemma 38. *The class of $(2P_2, \overline{P_2 + P_3})$-free atoms that contain an induced C_5 has bounded clique-width.*

Proof. Suppose G is a $(2P_2, \overline{P_2 + P_3})$-free atom containing an induced cycle C on five vertices, say $v_1, \ldots, v_5$ in that order. For $S \subseteq \{1, \ldots, 5\}$, let V_S be the set of vertices $x \in V(G) \setminus V(C)$ such that $N(x) \cap V(C) = \{v_i \mid i \in S\}$.

To simplify notation, in the following claims, subscripts on vertices and vertex sets should be interpreted modulo 5 and whenever possible we will write V_i instead of $V_{\{i\}}$, write $V_{i,j}$ instead of $V_{\{i,j\}}$, and so on.

Claim 39. *For $i \in \{1, \ldots, 5\}$, $V_i \cup V_{i,i+1} \cup V_{i-1,i,i+1}$ are empty.*

Proof of Claim. Suppose, for contradiction, that $x \in V_2 \cup V_{2,3} \cup V_{1,2,3}$. Then $G[x, v_2, v_4, v_5]$ is a $2P_2$, a contradiction. The claim follows by symmetry. $\diamond$

By Claim 39, the only non-empty sets V_S are those of the form $V_\emptyset$, $V_{i,i+2}$, $V_{i,i+1,i+3}$, $V_{i,i+1,i+2,i+3}$ and $V_{1,2,3,4,5}$. We now prove a sequence of claims.

Claim 40. *For $i \in \{1, \ldots, 5\}$, $V_\emptyset \cup V_{i,i+2}$ is independent.*

Proof of Claim. Suppose, for contradiction, that $x, y \in V_\emptyset \cup V_{1,3}$ are adjacent. Then $G[v_4, v_5, x, y]$ is a $2P_2$, a contradiction. The claim follows by symmetry. $\diamond$

Claim 41. *For $i \in \{1, \ldots, 5\}$, $|V_{i,i+1,i+3} \cup V_{i,i+1,i+2,i+3}| \leq 1$.*

Proof of Claim. Suppose, for contradiction that there are distinct vertices $x, y \in V_{1,2,4} \cup V_{1,2,3,4}$. Then $G[v_1, v_4, x, v_5, y]$ or $G[x, y, v_1, v_4, v_2]$ is a $\overline{P_2 + P_3}$ if x is adjacent or non-adjacent to y, respectively, a contradiction. The claim follows by symmetry. $\diamond$

[1]Let D be a prime atom in $\mathcal{G}$. As D is prime, D cannot contain both u and v. This means that D is split graph. By Lemma 37, as D is an atom, it must be a complete graph. As D is prime, this implies that $|V(D)| \leq 2$.

Claim 42. *For $i \in \{1, \ldots, 5\}$, there is at most one edge between $V_{i,i+2}$ and $V_{i,i-2}$.*

Proof of Claim. Suppose, for contradiction, that a vertex $x \in V_{1,3}$ has two neighbors $y, y' \in V_{1,4}$. By Claim 40, the sets $V_{1,3}$ and $V_{1,4}$ are independent. In particular, this means that y is non-adjacent to y'. Therefore $G[y, y', x, v_4, v_1]$ is a $\overline{P_2 + P_3}$, a contradiction. It follows that every vertex in $V_{1,3}$ has at most one neighbor in $V_{1,4}$. By symmetry, every vertex in $V_{1,4}$ has at most one neighbor in $V_{1,3}$ and so the edges between $V_{1,3}$ and $V_{1,4}$ form a matching. Since G is $2P_2$-free, it follows that there is at most one edge between $V_{1,3}$ and $V_{1,4}$. The claim follows by symmetry. ◇

Claim 43. *For $i \in \{1, \ldots, 5\}$, $V_{i,i+2}$ is complete to $V_{i-1,i+1} \cup V_{i+1,i+3}$.*

Proof of Claim. Suppose, for contradiction, that $x \in V_{1,3}$ is non-adjacent to $y \in V_{2,4}$. Then $G[x, v_1, y, v_4]$ is a $2P_2$, a contradiction. The claim follows by symmetry. ◇

Claim 44. *If $x \in V_{1,2,3,4,5}$, then x is complete to $V(G) \setminus \{x\}$. In particular, this implies that $V_{1,2,3,4,5}$ is a clique.*

Proof of Claim. Let $x \in V_{1,2,3,4,5}$ and suppose, for contradiction, that $y \in V(G) \setminus \{x\}$ is non-adjacent to x. Clearly $y \notin V(C)$. If $y \in V_{1,2,3,4,5}$, then $G[x, y, v_1, v_4, v_2]$ is a $\overline{P_2 + P_3}$, a contradiction. If y is adjacent to v_i and v_{i+2}, but not to v_{i+1} for some $i \in \{1, \ldots, 5\}$, then $G[v_i, v_{i+2}, v_{i+1}, y, x]$ is a $\overline{P_2 + P_3}$, a contradiction. By Claim 39, it follows that $y \in V_\emptyset$. Note that this implies that every vertex of $V_{1,2,3,4,5}$ is adjacent to every other vertex in $V(G) \setminus V_\emptyset$.

Since G is an atom, $N(y)$ cannot be a clique, and so it must contain two non-adjacent vertices, say u and v. By Claim 40, $u, v \notin V_\emptyset$ and for all $i \in \{1, \ldots, 5\}$, $u, v \notin V_{i,i+2}$. Since every vertex of $V_{1,2,3,4,5}$ is adjacent to every other vertex in $V(G) \setminus V_\emptyset$, neither u nor v is equal to x and, furthermore, x is adjacent to both u and v. By Claim 39, it follows that u and v must each have at least three neighbors in C. Therefore u and v must have a common neighbor in C; let v_i be such a common neighbor. Now $G[u, v, x, y, v_i]$ is a $\overline{P_2 + P_3}$, a contradiction. This completes the proof of the claim. ◇

We now show how to use a bounded number of vertex deletions, complementations and bipartite complementations to change G into an edgeless graph. First, by Claim 41, we can make $V_{i,i+1,i+3}$ and $V_{i,i+1,i+2,i+3}$ empty for all $i \in \{1, \ldots, 5\}$ by deleting at most five vertices. See Figure 5.1 for an illustration of the resulting graph. Next, by Claim 44 we can apply a bipartite complementation between $V_{1,2,3,4,5}$ and the rest of the graph to disconnect $G[V_{1,2,3,4,5}]$ from it. Next, by Claim 44 we can apply a complementation to $V_{1,2,3,4,5}$, which turns it into an independent set. Now, by Claim 39, the only other vertices remaining are those in C, those in $V_\emptyset$ and those in $V_{i,i+2}$ for $i \in \{1, \ldots, 5\}$. Next, by Claim 42, we can make $V_{i,i+2}$ anti-complete to $V_{i,i-2}$ for all $i \in \{1, \ldots, 5\}$ by deleting at most five vertices. By Claim 40, the only remaining edges are those between $V_{i-1,i+1} \cup \{v_i\}$ and $V_{i,i+2} \cup \{v_{i+1}\}$ for $i \in \{1, \ldots, 5\}$. By Claim 43 combined with the definition of $V_{i,i+2}$, we can apply a bipartite complementation between each of these pairs to remove all remaining edges of the graph. Thus, applying at most ten vertex deletions, six bipartite complementations and one complementation operation to G, we obtain an edgeless graph, which has clique-width 1. By Facts 1, 2 and 3, it follows that G has bounded clique-width. □

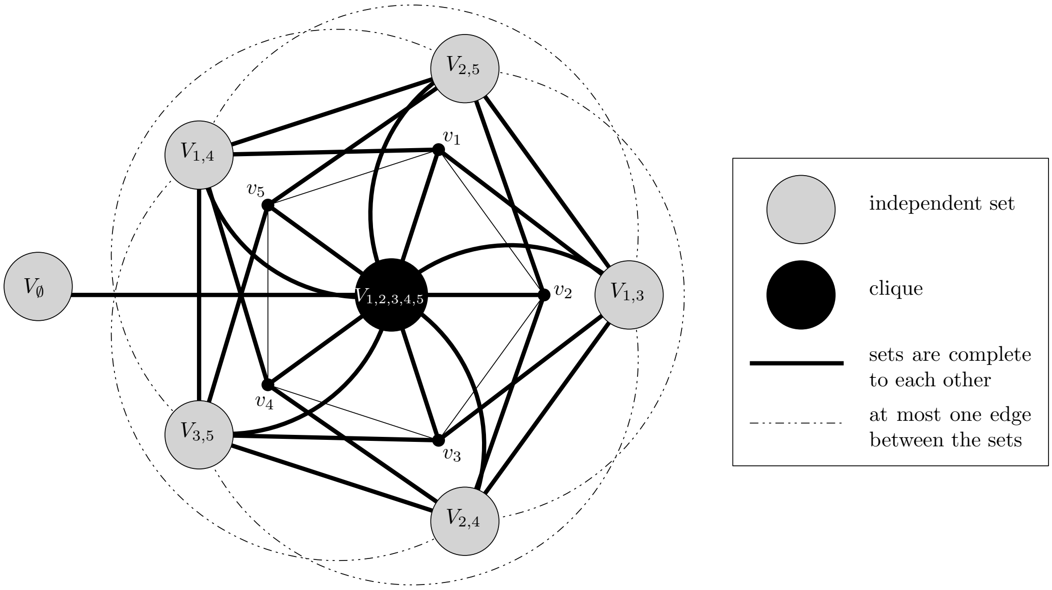


Figure 5.1 The configuration of the sets in the proof of Lemma 38 obtained after deleting the vertex sets $V_{i,i+1,i+3}$ and $V_{i,i+1,i+2,i+3}$ for $i \in \{1,2,3,4,5\}$. Note that vertices in $V_\emptyset$ now have no neighbors outside $V_{1,2,3,4,5}$.

Lemma 45. *The class of* $(2P_2, \overline{P_2+P_3})$*-free atoms that contain an induced* C_4 *has bounded clique-width.*

Proof. Suppose G is a $(2P_2, \overline{P_2+P_3})$-free atom containing an induced cycle C on four vertices, say $v_1, \ldots, v_4$ in that order. By Lemma 38, we may assume that G is C_5-free. For $S \subseteq \{1, \ldots, 4\}$, let V_S be the set of vertices $x \in V(G) \setminus V(C)$ such that $N(x) \cap V(C) = \{v_i \mid i \in S\}$.

To simplify notation, in the following claims, subscripts on vertices and vertex sets should be interpreted modulo 4 and whenever possible we will write V_i instead of $V_{\{i\}}$, write $V_{i,j}$ instead of $V_{\{i,j\}}$, and so on.

Claim 46. *For* $i \in \{1, \ldots, 4\}$*,* $V_{i,i+1,i+2}$ *is empty.*

Proof of Claim. Suppose, for contradiction, that $x \in V_{1,2,3}$. Then $G[v_1, v_3, v_2, v_4, x]$ is a $\overline{P_2+P_3}$, a contradiction. The claim follows by symmetry. ◇

See Figure 5.2 for an illustration of the remaining sets V_S that can be non-empty.

Claim 47. *For* $i \in \{1, \ldots, 4\}$*,* $V_\emptyset \cup V_i \cup V_{i+1} \cup V_{i,i+1}$ *is an independent set.*

Proof of Claim. Suppose, for contradiction, that $x, y \in V_\emptyset \cup V_1 \cup V_2 \cup V_{1,2}$ are adjacent. Then $G[x, y, v_3, v_4]$ is a $2P_2$, a contradiction. The claim follows by symmetry. ◇

Claim 48. *For* $i \in \{1, \ldots, 4\}$*,* $V_{i,i+1} \cup V_{i,i+2}$ *and* $V_{i,i+1} \cup V_{i+1,i+3}$ *are independent sets.*

Proof of Claim. Suppose, for contradiction, that $x, y \in V_{1,2} \cup V_{1,3}$ are adjacent. By Claim 47, x and y cannot both be in $V_{1,2}$, so assume without loss of generality that $x \in V_{1,3}$. Now $G[x, v_2, v_1, v_3, y]$ is a $\overline{P_2+P_3}$ (regardless of whether $y \in V_{1,2}$ or $y \in V_{1,3}$), a contradiction. The claim follows by symmetry. ◇

Claim 49. $G[V_{1,2,3,4}]$ *is* $(P_1 + P_2)$*-free and so it has bounded clique-width.*

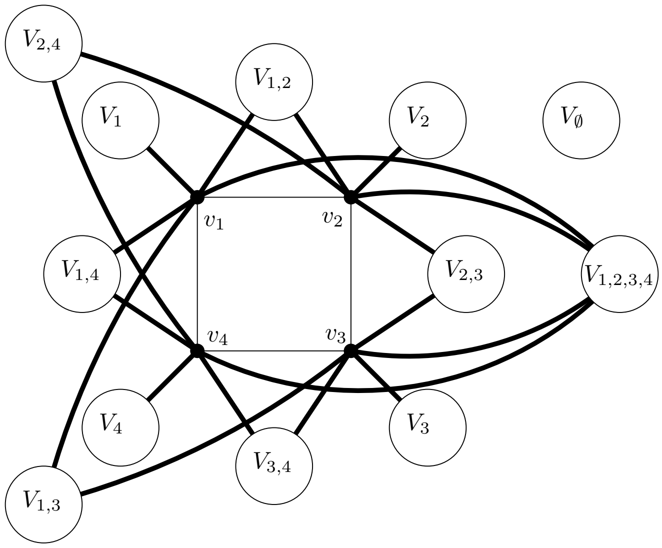


Figure 5.2 The possible non-empty sets V_S in the initial situation in the proof of Lemma 45. Edges between the different sets V_S are not drawn.

Proof of Claim. Suppose, for contradiction, that $x, y, y' \in V_{1,2,3,4}$ induce a $P_1 + P_2$ in G. Then $G[v_1, v_3, y, x, y']$ is a $\overline{P_2 + P_3}$, a contradiction. Therefore $G[V_{1,2,3,4}]$ is $(P_1 + P_2)$-free and so P_4-free, so it has bounded clique-width by Theorem 34. ◇

Claim 50. *For $i \in \{1, 2\}$, $V_{i,i+2}$ is complete to $V_{1,2,3,4}$.*

Proof of Claim. Suppose, for contradiction, that $x \in V_{1,3}$ is non-adjacent to $y \in V_{1,2,3,4}$. Then $G[v_1, v_3, v_2, x, y]$ is a $\overline{P_2 + P_3}$, a contradiction. The claim follows by symmetry. ◇

Claim 51. *For $i \in \{1, 2, 3, 4\}$ either $V_{i-1} \cup V_{i-1,i}$ or $V_{i,i+1} \cup V_{i+1}$ is empty.*

Proof of Claim. Suppose, for contradiction, that $x \in V_1 \cup V_{1,2}$ and $y \in V_{2,3} \cup V_3$. Then $G[v_1, x, y, v_3, v_4]$ is a C_5 or $G[x, v_1, y, v_3]$ is a $2P_2$ if x is adjacent or non-adjacent to y, respectively, a contradiction. The claim follows by symmetry. ◇

Claim 52. *If $x \in V_\emptyset$, then x has at least two neighbors in one of $V_{1,3}$ and $V_{2,4}$ and is anti-complete to the other. Furthermore, in this case x is complete to $V_{1,2,3,4}$.*

Proof of Claim. Suppose $x \in V_\emptyset$. Since G is not an atom, $N(x)$ cannot be a clique, and so must contain two non-adjacent vertices y, y'. By Claims 46 and 47, and the definition of $V_\emptyset$, it follows that $y, y' \in V_{1,3} \cup V_{2,4} \cup V_{1,2,3,4}$. If $y, y' \in V_{1,2,3,4}$, then $G[y, y', v_1, x, v_2]$ is a $\overline{P_2 + P_3}$, a contradiction. By Claim 50, $V_{1,2,3,4}$ is complete to $V_{1,3} \cup V_{2,4}$, so it follows that $y, y' \in V_{1,3} \cup V_{2,4}$. If $y \in V_{1,3}$ and $y' \in V_{2,4}$, then $G[v_1, v_2, y', x, y]$ is a C_5, a contradiction. It follows that $y, y' \in V_{1,3}$ or $y, y' \in V_{2,4}$.

Suppose $y, y' \in V_{1,3}$. If $z \in V_{2,4}$ is a neighbor of x, then z must be adjacent to y and y' (since, as shown above, x cannot have a pair of non-adjacent neighbors one of which is in $V_{1,3}$ and the other of which is in $V_{2,4}$), in which case $G[y, y', x, v_1, z]$ is a $\overline{P_2 + P_3}$, a contradiction. Therefore x cannot have a neighbor in $V_{2,4}$. If $z \in V_{1,2,3,4}$ is a non-neighbor of x, then z must be adjacent to y and y' by Claim 50, so $G[y, y', v_1, x, z]$ is a $\overline{P_2 + P_3}$, a contradiction. Therefore x is complete to $V_{1,2,3,4}$. The claim follows by symmetry. ◇

Claim 53. *For $i \in \{1,2\}$, $|V_{i,i+1} \cup V_{i+2,i+3}| \leq 2$.*

Proof of Claim. Suppose, for contradiction, that $|V_{1,2} \cup V_{3,4}| \geq 3$. First note that if $x \in V_{1,2}$, $y \in V_{3,4}$ are non-adjacent, then $G[v_1, x, v_3, y]$ is a $2P_2$, a contradiction. Therefore $V_{1,2}$ is complete to $V_{3,4}$. By Claim 47, both $V_{1,2}$ and $V_{3,4}$ are independent sets. If $x \in V_{1,2}$ and $y, y' \in V_{3,4}$, then $G[y, y', v_3, x, v_4]$ is a $\overline{P_2 + P_3}$, a contradiction. By symmetry, we conclude that either $V_{1,2}$ or $V_{3,4}$ is empty.

Suppose $V_{3,4}$ is empty, so $V_{1,2}$ contains at least three vertices and let $x \in V_{1,2}$ be such a vertex. Since G is an atom, $N(x)$ cannot be a clique, so x must have two neighbors y, y' that are non-adjacent. By Claims 46, 47, 48 and 51, and the definition of $V_{1,2}$, every neighbor of $x \in V_{1,2}$ lies in $\{v_1, v_2\} \cup V_{1,2,3,4}$. Since v_1 is complete to $\{v_2\} \cup V_{1,2,3,4}$ and v_2 is complete to $\{v_1\} \cup V_{1,2,3,4}$, it follows that $y, y' \in V_{1,2,3,4}$. Now $G[y, y', v_1, v_3, x]$ is a $\overline{P_2 + P_3}$, a contradiction. The claim follows by symmetry. ◇

Claim 54. *For $i \in \{1,2,3,4\}$, V_i is complete to $V_{1,2,3,4}$ and at most one vertex of $V_{i,i+2}$ has neighbors in V_i.*

Proof of Claim. Suppose $x \in V_1$. Since G is an atom, x must have two neighbors y, y' that are non-adjacent. By Claims 46, 47 and 51, and the definition of V_1, every neighbor of x lies in $\{v_1\} \cup V_{1,3} \cup V_{2,4} \cup V_{1,2,3,4}$. If $y, y' \in V_{1,3} \cup V_{1,2,3,4}$, then $G[y, y', v_1, v_3, x]$ is a $\overline{P_2 + P_3}$, a contradiction. The vertex v_1 is complete to $V_{1,3} \cup V_{1,2,3,4}$. Therefore without loss of generality, we may assume $y \in V_{2,4}$. Furthermore, note that $V_{1,3}$ is an independent set by Claim 48, so x has at most one neighbor in $V_{1,3}$. Since V_1 is an independent set by Claim 47, it follows that $G[V_1 \cup V_{1,3}]$ is a bipartite graph with parts V_1 and $V_{1,3}$. Since G is $2P_2$-free, it follows that no two vertices in V_1 can have different neighbors in $V_{1,3}$. Therefore at most one vertex of $V_{1,3}$ has a neighbor in V_1. Now if $z \in V_{1,2,3,4}$, then z is adjacent to y by Claim 50. If x is non-adjacent to z, then $G[v_1, y, v_2, x, z]$ is a $\overline{P_2 + P_3}$, a contradiction. We conclude that V_1 is complete to $V_{1,2,3,4}$. The claim follows by symmetry. ◇

We now proceed as follows. By Claim 46, the set $V_{1,2,3} \cup V_{2,3,4} \cup V_{1,3,4} \cup V_{1,2,4}$ is empty. By Claims 51 and 53, there are at most two vertices in $V_{1,2} \cup V_{2,3} \cup V_{3,4} \cup V_{1,4}$, so after doing at most two vertex deletions, we may assume these sets are empty (note that the resulting graph may no longer be an atom). Applying four further vertex deletions, we can remove the cycle C from G. By Claim 54, at most one vertex of $V_{1,3}$ (resp. $V_{2,4}$) has a neighbor in V_1 (resp. V_2). Therefore, applying at most two further vertex deletions, we may assume that $V_{1,3}$ is anti-complete to V_1 and $V_{2,4}$ is anti-complete to V_2. By Claim 51, we may assume without loss of generality that V_3 and V_4 are empty (see Figure 5.3 for an illustration of the resulting graph).

The remaining vertices of G all lie in $V_\emptyset \cup V_1 \cup V_2 \cup V_{1,3} \cup V_{2,4} \cup V_{1,2,3,4}$ and by Fact 1, it suffices to show that this modified graph has bounded clique-width. By Claims 50, 52 and 54, $V_{1,2,3,4}$ is complete to $V_\emptyset \cup V_1 \cup V_2 \cup V_{1,3} \cup V_{2,4}$, and so applying a bipartite complementation between these two sets disconnects $G[V_{1,2,3,4}]$ from the rest of the graph. By Claim 49, $G[V_{1,2,3,4}]$ has bounded clique-width, so by Fact 3, we may assume $V_{1,2,3,4}$ is empty.

By Claim 52, we can partition $V_\emptyset$ into the set $V_\emptyset^{1,3}$ of vertices that have neighbors in $V_{1,3}$ and the set $V_\emptyset^{2,4}$ of vertices that have neighbors in $V_{2,4}$. Now Claims 47 and 48 imply that $V_\emptyset^{2,4} \cup V_1 \cup V_{1,3}$ and $V_\emptyset^{1,3} \cup V_2 \cup V_{2,4}$ are independent sets (recall that $V_{1,3}$ is now anti-complete to V_1 and $V_{2,4}$ is now anti-complete to V_2), and so $G[V_\emptyset \cup V_1 \cup V_2 \cup V_{1,3} \cup V_{2,4}]$

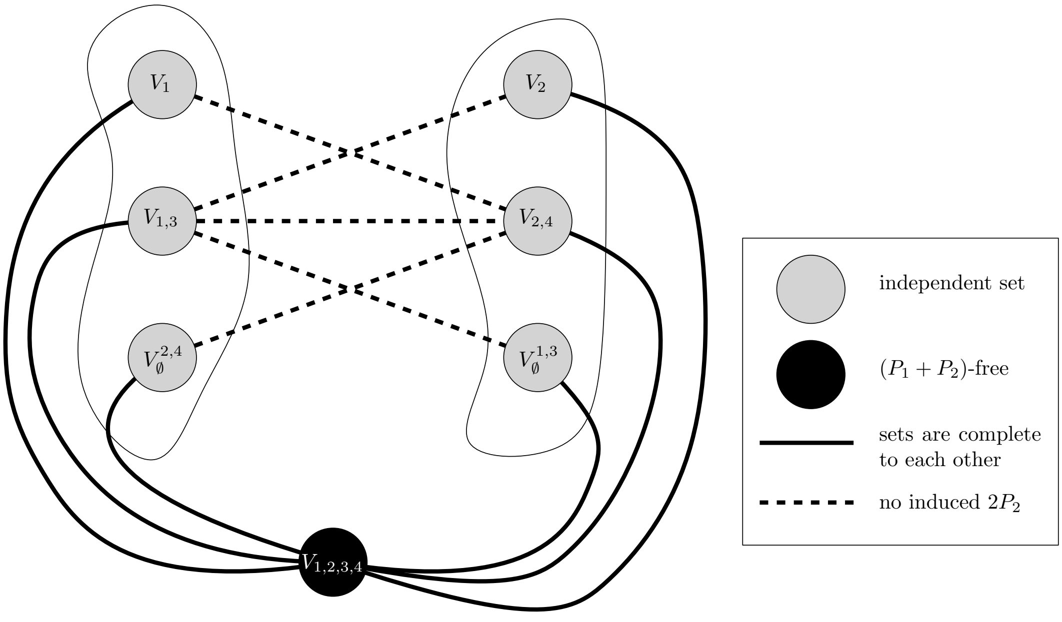


Figure 5.3 The configuration in the proof of Lemma 45 after deleting the at most six vertices in $V_{1,2} \cup V_{2,3} \cup V_{3,4} \cup V_{1,4} \cup V(C)$ along with at most one vertex in each of $V_{1,3}$ and $V_{2,4}$. The sets $V_\emptyset^{2,4} \cup V_1 \cup V_{1,3}$ and $V_\emptyset^{1,3} \cup V_2 \cup V_{2,4}$ are independent.

is a $2P_2$-free bipartite graph, so it is a bipartite chain graph by Lemma 35 and thus has bounded clique-width by Lemma 36. By Fact 1, this completes the proof. □

The class of split graphs is the class of $(C_4, C_5, 2P_2)$-free graphs. Since split graphs therefore form a subclass of the class of $(2P_2, \overline{P_2 + P_3})$-free graphs, and split graphs have unbounded clique-width [152], it follows that $(2P_2, \overline{P_2 + P_3})$-free graphs also have unbounded clique-width. Recall that by Lemma 37, split atoms are complete graphs and therefore have clique-width at most 2. The $(2P_2, \overline{P_2 + P_3})$-free atoms that are not split must therefore contain an induced C_4 or C_5. Applying Lemmas 38 and 45, we obtain Theorem 6 showing that the class of $(2P_2, \overline{P_2 + P_3})$-free atoms has bounded clique-width.

5.4 Clique-Width Summary for Bigenic Classes

In this section we present the state-of-art for boundedness of clique-width of general bigenic classes. We will use these results in the next section, where we prove our results on unboundedness of clique-width of atoms in bigenic classes.

Let H_1, H_2, H_3, H_4 be four graphs. Then the classes of (H_1, H_2)-free graphs and (H_3, H_4)-free graphs are said to be *equivalent* if the unordered pair H_3, H_4 can be obtained from the unordered pair H_1, H_2 by some combination of the operations: (i) complementing both graphs in the pair, and (ii) if one of the graphs in the pair is $3P_1$, replacing it with $P_1 + P_3$ or vice versa. If two classes are equivalent, then one of them has bounded clique-width if and only if the other one does [77].

Recall that the subdivided claw $S_{h,i,j}$, for $1 \leq h \leq i \leq j$ is the tree with one vertex x of degree 3 and exactly three leaves, which are of distance h, i and j from x, respectively. Also recall that $\mathcal{S}$ denotes the class of graphs every connected component of which is either a subdivided claw or a path. Moreover, recall that the paw is the graph $\overline{P_1 + P_3}$, the diamond is the graph $\overline{2P_1 + P_2}$ and the gem is the graph $\overline{P_1 + P_4}$.

Theorem 55 ([71])**.** *Let $\mathcal{G}$ be a class of graphs defined by two forbidden induced subgraphs. Then:*

1. *$\mathcal{G}$ has bounded clique-width if it is equivalent to a class of (H_1, H_2)-free graphs such that one of the following holds:*
   - *(i)* H_1 *or* $H_2 \subseteq_i P_4$
   - *(ii)* $H_1 = K_s$ *and* $H_2 = tP_1$ *for some* $s, t \geq 1$
   - *(iii)* $H_1 \subseteq_i \text{paw}$ *and* $H_2 \subseteq_i K_{1,3}+3P_1,\ K_{1,3}+P_2,\ P_1+P_2+P_3,\ P_1+P_5,\ P_1+S_{1,1,2},\ P_2+P_4,\ P_6,\ S_{1,1,3}$ *or* $S_{1,2,2}$
   - *(iv)* $H_1 \subseteq_i \text{diamond}$ *and* $H_2 \subseteq_i P_1 + 2P_2,\ 3P_1 + P_2$ *or* $P_2 + P_3$
   - *(v)* $H_1 \subseteq_i \text{gem}$ *and* $H_2 \subseteq_i P_1 + P_4$ *or* P_5
   - *(vi)* $H_1 \subseteq_i K_3 + P_1$ *and* $H_2 \subseteq_i K_{1,3}$*, or*
   - *(vii)* $H_1 \subseteq_i \overline{2P_1 + P_3}$ *and* $H_2 \subseteq_i 2P_1 + P_3$.

2. *$\mathcal{G}$ has unbounded clique-width if it is equivalent to a class of (H_1, H_2)-free graphs such that one of the following holds:*
   - *(i)* $H_1 \notin \mathcal{S}$ *and* $H_2 \notin \mathcal{S}$
   - *(ii)* $H_1 \notin \overline{\mathcal{S}}$ *and* $H_2 \notin \overline{\mathcal{S}}$
   - *(iii)* $H_1 \supseteq_i K_3 + P_1$ *or* C_4 *and* $H_2 \supseteq_i 4P_1$ *or* $2P_2$
   - *(iv)* $H_1 \supseteq_i \text{diamond}$ *and* $H_2 \supseteq_i K_{1,3},\ 5P_1,\ P_2 + P_4$ *or* P_6
   - *(v)* $H_1 \supseteq_i K_3$ *and* $H_2 \supseteq_i 2P_1 + 2P_2,\ 2P_1 + P_4,\ 4P_1 + P_2,\ 3P_2$ *or* $2P_3$
   - *(vi)* $H_1 \supseteq_i K_4$ *and* $H_2 \supseteq_i P_1 + P_4$ *or* $3P_1 + P_2$*, or*
   - *(vii)* $H_1 \supseteq_i \text{gem}$ *and* $H_2 \supseteq_i P_1 + 2P_2$.

As already mentioned, Theorem 55 does not cover five (non-equivalent) cases (see also Open Problem 2, where these open cases are marked with a *).

Open Problem 1. *Does the class of (H_1, H_2)-free graphs have bounded or unbounded clique-width when:*

- *(i)* $H_1 = K_3$ *and* $H_2 \in \{P_1 + S_{1,1,3}, S_{1,2,3}\}$
- *(ii)* $H_1 = \text{diamond}$ *and* $H_2 \in \{P_1 + P_2 + P_3, P_1 + P_5\}$
- *(iii)* $H_1 = \text{gem}$ *and* $H_2 = P_2 + P_3$.

5.5 Atoms of Unbounded Clique-Width

In this section we show our results for pairs (H_1, H_2), for which the class of (H_1, H_2)-free atoms has unbounded clique-width. We start by giving a number of known and new lemmas, some of which have wider applicability.

Lemma 56 ([77])**.** *For $m \geq 0$ and $n > m+1$ the clique-width of a graph G is at least $\lfloor \frac{n-1}{m+1} \rfloor + 1$ if $V(G)$ has a partition into sets $V_{i,j}$ ($i, j \in \{0, \ldots, n\}$) with the following properties:*

1. $|V_{i,0}| \leq 1$ *for all* $i \in \{1, \ldots, n\}$
2. $|V_{0,j}| \leq 1$ *for all* $j \in \{1, \ldots, n\}$
3. $|V_{i,j}| \geq 1$ *for all* $i, j \in \{1, \ldots, n\}$
4. $G[\cup_{j=0}^{n} V_{i,j}]$ *is connected for all* $i \in \{1, \ldots, n\}$
5. $G[\cup_{i=0}^{n} V_{i,j}]$ *is connected for all* $j \in \{1, \ldots, n\}$
6. *for* $i, j, k \in \{1, \ldots, n\}$*, if a vertex of* $V_{k,0}$ *is adjacent to a vertex of* $V_{i,j}$ *then* $i \leq k$
7. *for* $i, j, k \in \{1, \ldots, n\}$*, if a vertex of* $V_{0,k}$ *is adjacent to a vertex of* $V_{i,j}$ *then* $j \leq k$*, and*
8. *for* $i, j, k, \ell \in \{1, \ldots, n\}$*, if a vertex of* $V_{i,j}$ *is adjacent to a vertex of* $V_{k,\ell}$ *then* $|k - i| \leq m$ *and* $|\ell - j| \leq m$*.*

The next lemma concerns walls. We do not formally define the wall, but instead we refer to Figure 5.4, in which three examples of walls of different heights are depicted; see, for example, [61] for a formal definition.

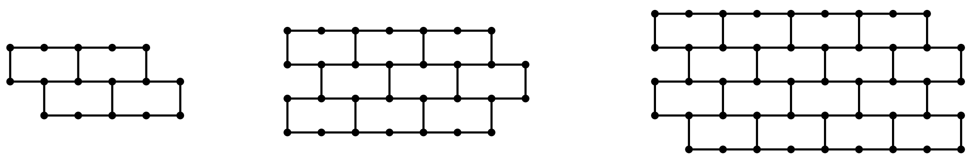


Figure 5.4 Walls of height 2, 3 and 4, respectively.

Lemma 57 ([148])**.** *For any constant $k \geq 0$, the class of k-subdivided walls has unbounded clique-width.*

Lemma 58. *Let H_1, H_2 be graphs. If $H_1, H_2 \notin \mathcal{S}$ or $\overline{H_1}, \overline{H_2} \notin \mathcal{S}$ then the class of (H_1, H_2)-free atoms has unbounded clique-width.*

Proof. Let $k = \max(|V(H_1)|, |V(H_2)|)$. Let H be a k-subdivided wall of height at least 2 (see Figure 5.4). Note that the clique-width of k-subdivided walls is unbounded by Lemma 57. By Fact 2, it follows that the clique-width of graphs of the form $\overline{H}$ is also unbounded.

We claim that if $H_1, H_2 \notin \mathcal{S}$ then H is (H_1, H_2)-free. Let $i \in \{1,2\}$. It is easy to verify that if H_i contains a cycle, then H is H_i-free (due to the choice of k). Similarly, if H_i contains an induced tree with two vertices of degree at least 3 or a vertex of degree at least 4, then H is H_i-free. Therefore, if H_i is an induced subgraph of H, then H_i is a forest and every component of H_i must be a tree in which at most one vertex has degree 3 and all other vertices have degree at most 2. In other words, if H_i is an induced subgraph of H, then $H_i \in \mathcal{S}$. We conclude that if $H_1, H_2 \notin \mathcal{S}$, then H is (H_1, H_2)-free. This also implies that if $\overline{H_1}, \overline{H_2} \notin \mathcal{S}$, then $\overline{H}$ is (H_1, H_2)-free.

It remains to show that H and $\overline{H}$ are atoms. Indeed, H is a bipartite graph, so every clique cut-set consists of at most two vertices; it is easy to verify that there is no vertex whose removal disconnects H and no edge such that removing both of its end-vertices disconnects H. Therefore H is indeed an atom.

Now, $\overline{H}$ is a co-bipartite graph, so it can be partitioned into two cliques A and B. Note that $|A|, |B| > 12$ by construction. Suppose, for contradiction, that X is a clique cut-set in $\overline{H}$. Let $Y = V(\overline{H}) \setminus X$ and note that $\overline{H}[Y]$ is disconnected, so it contains two vertices a, b that are non-adjacent. Since A is a clique and B is a clique, we may assume $a \in A$ and $b \in B$. Now Y cannot contain vertices $a' \in A$, $b' \in B$ that are adjacent in $\overline{H}$, as in that case $\{a', b'\}$ would dominate $\overline{H}$, contradicting the assumption that $\overline{H}[Y]$ is disconnected. In H every vertex has either two or three neighbors, so in $\overline{H}$ every vertex has either two or three non-neighbors. Since $a \in A \cap Y$, there can be at most three vertices in $B \cap Y$ and similarly, there can be at most three vertices in $A \cap Y$. Since every vertex in $B \cap Y$ has at most three non-neighbors in A, it follows that at most nine vertices of A have non-neighbors in $B \cap Y$. Since $|A| > 12 \geq 9 + |A \cap Y|$, there must be a vertex in $a' \in A \setminus Y = A \cap X$ that has no non-neighbors in $B \cap Y$ and therefore has a non-neighbor $b' \in B \setminus Y = B \cup X$. This contradicts the fact that X is a clique in $\overline{H}$. Therefore $\overline{H}$ is indeed an atom. □

Lemma 59. *Let $\mathcal{H}$ be a set of graphs such that no graph in $\mathcal{H}$ contains a pair of vertices that are false twins. Then the class of $\mathcal{H}$-free atoms has bounded clique-width if and only if the class of $\mathcal{H}$-free graphs does.*

Proof. Clearly, if the class of $\mathcal{H}$-free graphs has bounded clique-width, then the class of $\mathcal{H}$-free atoms does. Now suppose that the class of $\mathcal{H}$-free graphs has unbounded clique-width. Let $\mathcal{F}$ be the class of connected $\mathcal{H}$-free graphs on at least two vertices. Since the clique-width of a graph is equal to the maximum of the clique-widths of its components, it follows that $\mathcal{F}$ has unbounded clique-width. For every graph $F \in \mathcal{F}$, we construct the graph F', which has vertex set $V(F') = \{v, v' \mid v \in V(F)\}$ and edge set $E(F') = \{uv, uv', u'v, u'v' \mid uv \in E(F)\}$. So, for every $v \in V(F)$ we have introduced a new vertex v' such that v and v' are false twins in F'. Let $\mathcal{F}'$ be the set of such graphs F'. Since for every $F \in \mathcal{F}$, the graph F' contains F as an induced subgraph, it follows that $\mathcal{F}'$ has unbounded clique-width.

We claim that every graph in $\mathcal{F}'$ is an atom. Indeed, suppose, for contradiction, that X is a clique cut-set of a graph $F' \in \mathcal{F}'$. Since for every $v \in V(F)$, v is non-adjacent to v' in F', it follows that at most one of v and v' is in X. Since v and v' are false twins in F' we may replace all vertices $v \in X \cap V(F)$ by their false twins v' and the resulting set X' will still be a clique cut-set. By construction, the graph F is connected and every vertex in $V(F') \setminus X'$ has a neighbor in $V(F)$ in the graph $F' \setminus X'$. Therefore $F' \setminus X'$ is connected, a contradiction. It follows that every graph in $\mathcal{F}'$ is indeed an atom.

It remains to show that the graphs in $\mathcal{F}'$ are $\mathcal{H}$-free. Indeed, suppose, for contradiction, that $H \in \mathcal{H}$ is an induced subgraph of $F' \in \mathcal{F}'$. Since for every $v \in V(F)$, the vertices v and v' are false twins in F', and H does not have a pair of false twins, it follows that at most one of v and v' is in the induced copy of H found in F'. Furthermore, if v' is in this induced copy, then we can replace it by v. Thus we find that there is an induced copy of H in F' all of whose vertices lie in $V(F)$. Therefore H is an induced subgraph of F. This is a contradiction as $F \in \mathcal{F}$ and the graphs in $\mathcal{F}$ are $\mathcal{H}$-free. We have therefore shown that the graphs in $\mathcal{F}'$ are $\mathcal{H}$-free atoms and that $\mathcal{F}'$ has unbounded clique-width. This completes the proof. □

Observe that the condition in the following lemma holds if and only if for every graph $H \in \mathcal{H}$, the graph $\overline{H}$ does not have a component isomorphic to P_1 or P_2.

Lemma 60. *Let $\mathcal{H}$ be a set of graphs such that no graph in $\mathcal{H}$ contains a dominating vertex and no graph in $\mathcal{H}$ contains a pair of non-adjacent vertices that are complete to the remainder of the graph. Then the class of $\mathcal{H}$-free atoms has bounded clique-width if and only if the class of $\mathcal{H}$-free graphs does.*

Proof. Clearly, if the class of $\mathcal{H}$-free graphs has bounded clique-width, then the class of $\mathcal{H}$-free atoms does. Now suppose that the class of $\mathcal{H}$-free graphs has unbounded clique-width. Let $\mathcal{F}$ be the class of $\mathcal{H}$-free graphs that contain at least one non-edge. Since complete graphs have clique-width at most 2 and the class of $\mathcal{H}$-free graphs has unbounded clique-width, it follows that $\mathcal{F}$ has unbounded clique-width. For every graph $F \in \mathcal{F}$, we construct the graph F' by adding two new vertices x, x' and adding edges to make $\{x, x'\}$ complete to the remainder of the graph (note that x is non-adjacent to x' in F'). Let $\mathcal{F}'$ be the set of such graphs F'. Since for every $F \in \mathcal{F}$, the graph F' contains F as an induced subgraph, it follows that $\mathcal{F}'$ has unbounded clique-width.

We claim that every graph in $\mathcal{F}'$ is an atom. Suppose, for contradiction, that $F' \in \mathcal{F}'$ has a clique cut-set X. Since x and x' are non-adjacent, it follows that either x or x' are not in X; since x and x' are false twins, we may assume $x \notin X$. Since F is not a complete graph, there must be a vertex $y \in V(F) \setminus X$. Since x is complete to $V(F) \setminus X$ in $F' \setminus X$, every vertex of $V(F) \setminus X$ is in the same component of $F' \setminus X$ as x. Since y is complete to $\{x, x'\} \setminus X$ in $F' \setminus X$, every vertex of $\{x, x'\} \setminus X$ is in the same component of $F' \setminus X$ as y. Therefore $F' \setminus X$ is connected, a contradiction. It follows that every graph in $\mathcal{F}'$ is indeed an atom.

It remains to show that the graphs in $\mathcal{F}'$ are $\mathcal{H}$-free. Indeed, suppose, for contradiction, that $H \in \mathcal{H}$ is an induced subgraph of $F' \in \mathcal{F}'$. Since H does not contain a pair of non-adjacent vertices that are complete to the rest of the graph, this induced copy of H in F' cannot contain both x and x'. Since H does not have a dominating vertex, the induced copy of H in F' cannot contain exactly one of x and x'. Therefore the induced copy of H in F' must consist of only vertices in $V(F)$. Therefore H is an induced subgraph of F. This is a contradiction as $F \in \mathcal{F}$ and the graphs in $\mathcal{F}$ are $\mathcal{H}$-free. We have therefore shown that the graphs in $\mathcal{F}'$ are $\mathcal{H}$-free atoms and that $\mathcal{F}'$ has unbounded clique-width. This completes the proof. □

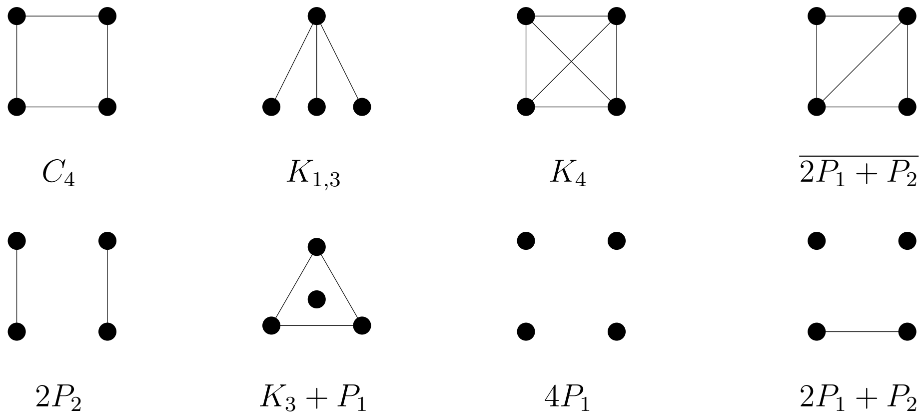


Figure 5.5 The forbidden induced subgraphs for the classes of $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$-free graphs and $(2P_2, K_3 + P_1, 4P_1, 2P_1 + P_2)$-free graphs mentioned in Lemma 61.

Lemma 61. *The class of* $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$*-free atoms and the class of* $(2P_2, K_3+P_1, 4P_1, 2P_1 + P_2)$*-free atoms have unbounded clique-width (see Figure 5.5 for illustrations of the forbidden induced subgraphs).*

Proof. Brandstädt et al. [31, Theorem 10(ii)] constructed a family of graphs H_n that have unbounded clique-width and are $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$-free. The graph H_n is constructed from the 1-subdivided $n \times n$ grid by adding new edges incident to the vertices added by the subdivision as follows: in each cell of the subdivided grid, the left vertex added by the subdivision is made adjacent to the top one, and the bottom vertex added by the subdivision is made adjacent to the right one (see also Figure 5.6 or see [31, Section 6.2] for a formal definition). However, the graph H_n has clique cut-sets, so it is not an atom. On the other hand, since the class of $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$-free graphs has unbounded clique-width, Fact 2 implies that the class of $(2P_2, K_3 + P_1, 4P_1, 2P_1 + P_2)$-free graphs has unbounded clique-width. We observe that every graph in $\{2P_2, K_3 + P_1, 4P_1, 2P_1 + P_2\}$ has no dominating vertex and no two non-adjacent vertices that are complete to the remainder of the graph. Therefore, by Lemma 60, the class of $(2P_2, K_3 + P_1, 4P_1, 2P_1 + P_2)$-free atoms has unbounded clique-width.

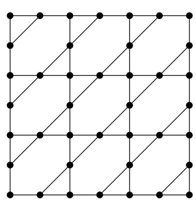


Figure 5.6 The graph H_n from the proof of Lemma 61 ($n = 4$ shown).

We now prove that the class of $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$-free atoms has unbounded clique-width. Consider a wall of height $k \geq 2$ and let J_k be its line graph. It is easy to verify that for every k, the graph H_n contains J_k as an induced subgraph if n is sufficiently large. Similarly, for every n, the graph J_k contains H_n as an induced subgraph if k is sufficiently large. Therefore, by [31, Theorem 10(ii)], the graph J_k is $(C_4, K_{1,3}, K_4, \overline{2P_1 + P_2})$-free and this family of graphs has unbounded clique-width (the former can also be seen by inspection and latter can also be seen by using Lemma 56). Every clique in J_k contains at most three vertices and it is easy to verify that J_k does not contain a clique cut-set on at most three vertices, so J_k is an atom. This completes the proof. □

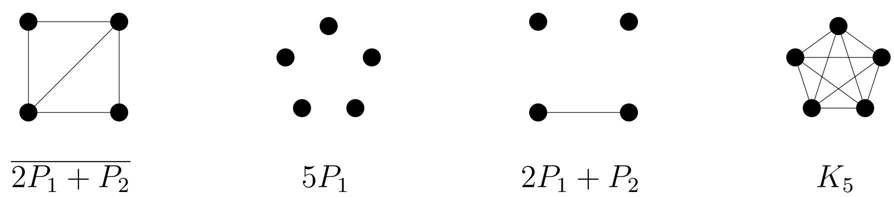


Figure 5.7 The forbidden induced subgraphs for the classes of $(\overline{2P_1 + P_2}, 5P_1)$-free graphs and $(2P_1 + P_2, K_5)$-free graphs mentioned in Lemma 62.

Lemma 62. *The class of $(\overline{2P_1 + P_2}, 5P_1)$-free atoms and the class of $(2P_1 + P_2, K_5)$-free atoms has unbounded clique-width (see Figure 5.7 for illustrations of the forbidden induced subgraphs).*

Proof. We use the construction from [69], which was used to show that $(\overline{2P_1 + P_2}, 5P_1)$-free graphs have unbounded clique-width. Consider a wall of height $2n + 1$ for some $n \geq 2$. Colour the vertices on the top row with colours $1, 2, 3, 4, 1, 2, 3, 4, \ldots$ and on the next row with colours $3, 4, 1, 2, 3, 4, 1, 2, \ldots$, then alternate these colorings on the following rows, so that no vertex has two neighbors that have the same colour (see also Figure 5.8). Add edges to make each colour class into a clique and let G_n be the resulting graph. Now G_n is $(\overline{2P_1 + P_2}, 5P_1)$-free and the family of such graphs had unbounded clique-width [69] (the former can also be seen by inspection and the latter follows from combining Lemma 56 with Fact 2). By Fact 2, the family of graphs $\overline{G_n}$ also has unbounded clique-width.

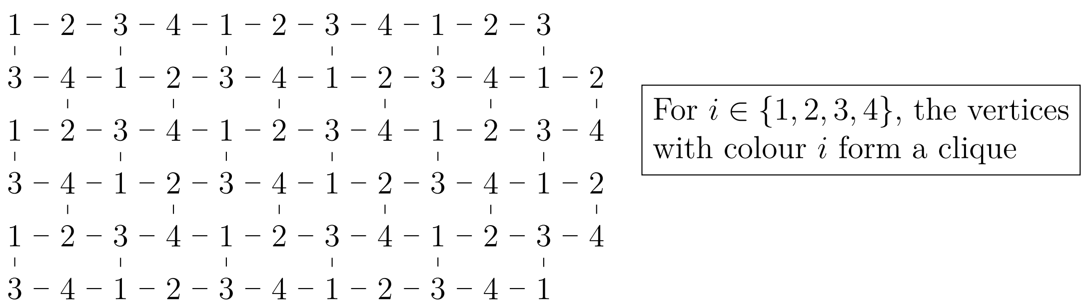


Figure 5.8 The graph G_n from the proof of Lemma 62 ($n = 2$ shown).

It remains to show that G_n and $\overline{G_n}$ are atoms. Let V_i be the set of vertices with colour i. Suppose, for contradiction, that G_n has a clique cut-set X. If $X \subseteq V_i$ for some $i \in \{1, 2, 3, 4\}$, then all vertices of $G_n \setminus V_i$ are in the same component of $G_n \setminus X$. Since every vertex in V_i has at least one neighbor outside of V_i, it follows that every vertex of $G_n \setminus X$ is in the same component of $G_n \setminus X$ in this case, a contradiction. We may therefore assume that X contains vertices in at least two sets V_i. By construction, each vertex in a set V_i has at most one neighbor in each V_j for $j \in \{1,2,3,4\} \setminus \{i\}$. Therefore X has at most one vertex in each V_i. Therefore, there must be a vertex in $V_1 \setminus X$ that has a neighbor in each of $V_2 \setminus X$, $V_3 \setminus X$ and $V_4 \setminus X$. Since each set V_i is a clique, it follows that $G_n \setminus X$ is connected. This contradiction implies that G_n is indeed an atom. Now suppose, for contradiction, that $\overline{G_n}$ has a clique cut-set X. Since $V_1, \ldots, V_4$ are independent sets in $\overline{G_n}$, X contains at most one vertex of any V_i. Since in $\overline{G_n}$ every vertex of V_i has at most one non-neighbor in each V_j for $j \in \{1,2,3,4\} \setminus \{i\}$, it follows that $\overline{G_n} \setminus X$ must be connected. This contradiction implies that $\overline{G_n}$ is indeed an atom. □

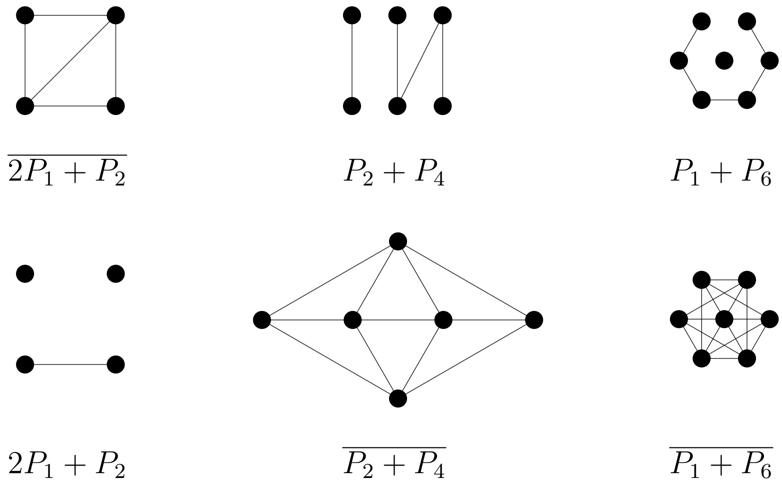


Figure 5.9 The forbidden induced subgraphs for the classes of $(\overline{2P_1 + P_2}, P_2 + P_4, P_1 + P_6)$-free graphs and $(2P_1 + P_2, \overline{P_2 + P_4}, \overline{P_1 + P_6})$-free graphs mentioned in Lemma 63.

Lemma 63. *The class of* $(\overline{2P_1 + P_2}, P_2 + P_4, P_1 + P_6)$*-free and the class of* $(2P_1 + P_2, \overline{P_2 + P_4}, \overline{P_1 + P_6})$*-free atoms have unbounded clique-width (see Figure 5.9 for illustrations of the forbidden induced subgraphs).*

Proof. We modify the construction of the graph G_n, which was used in [70] to prove that $(\overline{2P_1 + P_2}, P_2 + P_4)$-free graphs have unbounded clique-width. Consider a wall of height $n \geq 2$. A wall is a bipartite graph; let A and C be the two sets in its bipartition. Consider a 1-subdivision of the wall and let B be the set of vertices introduced by the subdivision. Finally, we add edges to make A complete to C. Let G_n be the resulting graph. Then G_n is $(\overline{2P_1 + P_2}, P_2 + P_4)$-free and the family of such graphs G_n has unbounded clique-width [70] (the former also follows by inspection and the latter follows from combining Lemma 57 with Fact 3). Let H_n be the graph obtained from G_n by adding a vertex x complete to B, see Figure 5.10. Since H_n contains G_n as an induced subgraph, the family of graphs H_n has unbounded clique-width.

Now G_n is $(\overline{2P_1 + P_2}, P_2 + P_4)$-free, so if H_n contains an induced copy of $\overline{2P_1 + P_2}$ or $P_2 + P_4$, then one of its vertices must be x. The neighborhood of x in H_n is B, which is an independent set. Every vertex of $\overline{2P_1 + P_2}$ has two neighbors that are adjacent to each other, so H_n is $\overline{2P_1 + P_2}$-free. Suppose, for contradiction, that H_n contains an induced $P_2 + P_4$, say with vertex set Y. As observed above, $x \in Y$. Now x has either one or two neighbors in $H_n[Y]$. If x has one neighbor in $H_n[Y]$, then this neighbor must

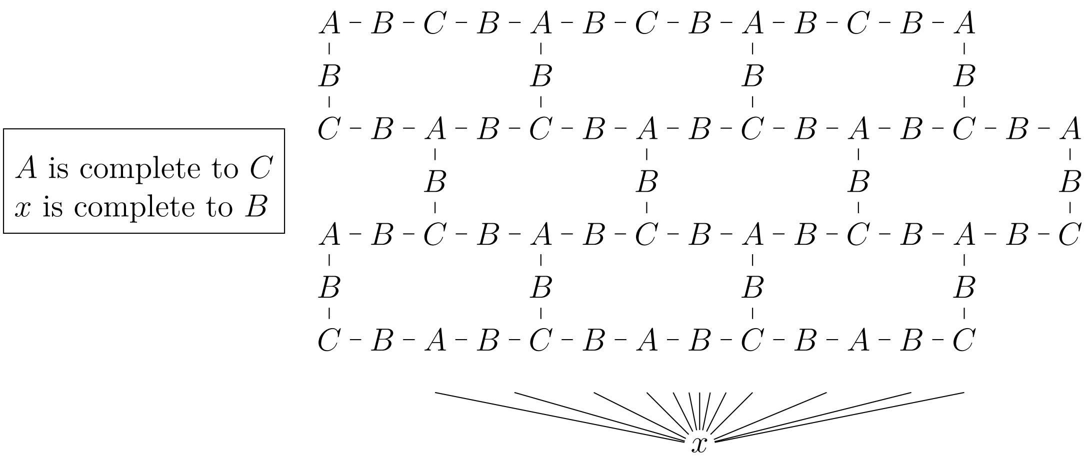


Figure 5.10 The graph H_n from the proof of Lemma 63 ($n = 3$ shown). Vertices are denoted A, B or C if they are in the corresponding set.

be in B, and then there can be no other vertices in $B \cap Y$, so $|(A \cup C) \cap Y| = 4$, but $H_n[A \cup C]$ is a complete bipartite graph, so $H_n[(A \cup C) \cap Y]$ is isomorphic to $C_4, K_{1,3}$ or $4P_1$, contradicting the fact that $P_2 + P_4$ is $(C_4, K_{1,3}, 4P_1)$-free. Therefore x has two neighbors in $H_n[Y]$, so it is in the P_4 component of $H_n[Y]$. These two neighbors of x must be in B, so the P_4 component containing x must contain a vertex of A or C and the remaining P_2 component of $H_n[Y]$ must lie in $A \cup C$. Since $H_n[A \cup C]$ is a complete bipartite graph, it follows that there is an edge between the P_2 component and the P_4 component. This contradiction implies that H_n is indeed $(P_2 + P_4)$-free.

Suppose, for contradiction, that H_n contains an induced $P_1 + P_6$, say with vertex set $\{v\} \cup Y$ where v is the vertex in the P_1 component and Y is the vertex set of the P_6 component. If there are three vertices in $B \cap Y$ and $x \in Y$, then $H_n[Y]$ contains an induced $K_{1,3}$, a contradiction. Note that P_6 has two vertices of degree 1 and four vertices of degree 2, but every vertex in B has only two neighbors apart from x: one in each of A and C. Therefore, if there are three vertices in $B \cap Y$, then one of these vertices b must have neighbors $a \in A \cap Y$ and $c \in C \cap Y$, in which case $H_n[a, b, c]$ is a K_3, a contradiction. We conclude that there are at most two vertices in $B \cap Y$. If $x \notin Y$ then there are at least four vertices in $(A \cup C) \cap Y$, contradicting the fact that P_6 is $(C_4, K_{1,3}, 4P_1)$-free. Therefore, $x \in Y$, and so $v \in A \cup C$ (say A) because x is complete to B. Since $\{x\} \cup A$ is independent and P_6 is $4P_1$-free, it follows that $|A \cap Y| \leq 2$, and so there is at least one vertex $c \in C \cap Y$. But c is complete to A, so it is adjacent to v, a contradiction. This contradiction implies that H_n is indeed $(P_1 + P_6)$-free.

It remains to show that H_n and $\overline{H_n}$ are atoms. Suppose, for contradiction, that H_n contains a clique cut-set X. If $x \in X$ then X contains at most one additional vertex, which must lie in B; it is easy to verify that $H_n \setminus X$ is connected in this case. We may therefore assume that $x \notin X$. Since A, B and C are independent sets, X contains at most one vertex in each of these sets. Since $x \notin X$, and x is complete to B, all vertices of $B \setminus X$ are in the same component of $H_n \setminus X$. Since every vertex of B has a neighbor in A and C, there must be a vertex in $B \setminus X$ that has neighbors in both $A \setminus X$ and $C \setminus X$. Since A is complete to C, it follows that every vertex in $V(H_n) \setminus X$ is in the same component of $H_n \setminus X$. This contradiction implies that H_n is indeed an atom. Now suppose, for contradiction, that $\overline{H_n}$ contains a clique cut-set X. Since A is anti-complete to C in $\overline{H_n}$, X cannot contain vertices in both A and C; by symmetry we may assume that X does not contain any vertices of C. Now C is a clique and, since every vertex of B has a neighbor in C, every vertex in $(B \cup C) \setminus X$ is in the same component of $\overline{H_n} \setminus X$. If $x \notin X$, then every vertex in $A \setminus X$ is adjacent to x, which is complete to C, so every vertex in $V(\overline{H_n}) \setminus X$ is in the same component of $\overline{H_n} \setminus X$, a contradiction. We may therefore assume that $x \in X$. Then no vertex of B is in X, so $X \subseteq A \cup \{x\}$. Since every vertex of A has a neighbor in B, it follows that every vertex of A has a neighbor in $B \setminus X = B$. Therefore every vertex of $V(\overline{H_n}) \setminus X$ is in the same component of $\overline{H_n} \setminus X$. This contradiction implies that $\overline{H_n}$ is indeed an atom. □

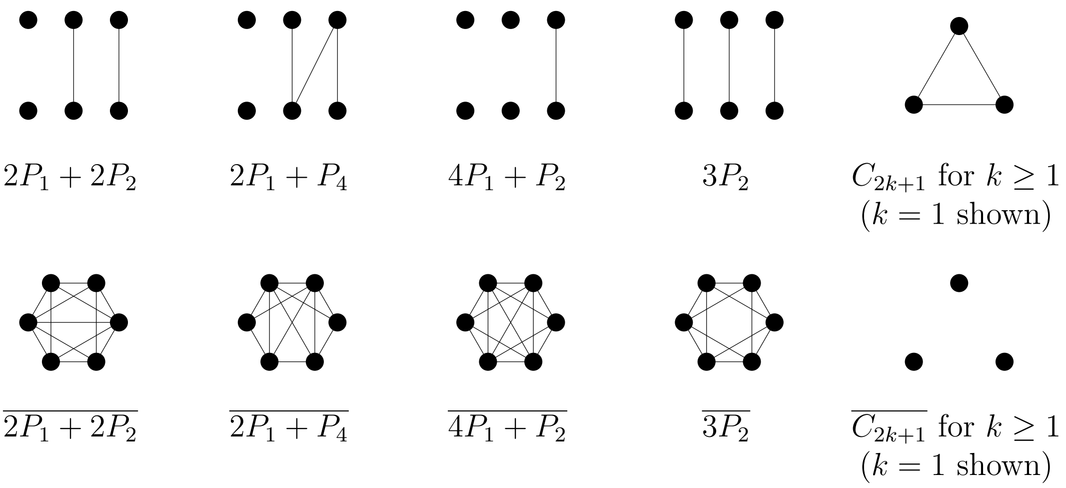


Figure 5.11 The forbidden induced subgraphs for the classes of $(2P_1+2P_2, 2P_1+P_4, 4P_1+P_2, 3P_2)$-free bipartite graphs and $(\overline{2P_1 + 2P_2}, \overline{2P_1 + P_4}, \overline{4P_1 + P_2}, \overline{3P_2})$-free co-bipartite graphs mentioned in Lemma 64.

Lemma 64. *The class of* $(2P_1 + 2P_2, 2P_1 + P_4, 4P_1 + P_2, 3P_2)$*-free bipartite atoms and the class of* $(\overline{2P_1 + 2P_2}, \overline{2P_1 + P_4}, \overline{4P_1 + P_2}, \overline{3P_2})$*-free co-bipartite atoms have unbounded clique-width (see Figure 5.11 for illustrations of the forbidden induced subgraphs).*

Proof. Let H_n be a 1-subdivided wall of height $n \geq 2$ and note that the class of such graphs has unbounded clique-width by Lemma 57. Note that H_n is connected and bipartite, say with parts V_1 and V_2. Let G_n be the graph obtained from H_n by applying a bipartite complementation between V_1 and V_2. By Fact 3, the family of such graphs G_n also has unbounded clique-width and by Fact 2 the family of graphs $\overline{G_n}$ also has unbounded clique-width (see also Figure 5.12). Now G_n is a $(2P_1 + 2P_2, 2P_1 + P_4, 4P_1 + P_2, 3P_2)$-free bipartite graph (by inspection, or see e.g. [76, 149]).

It remains to show that G_n and $\overline{G_n}$ are atoms. Suppose, for contradiction, that X is a clique cut-set of G_n. Since V_1 and V_2 are independent, X contains at most one vertex from each of these sets. Since every vertex of V_1 has at most three non-neighbors in V_2 and vice versa, it follows that $G_n \setminus X$ is connected. This contradiction shows that G_n is indeed an atom. Now suppose, for contradiction, that X is a clique cut-set of $\overline{G_n}$.

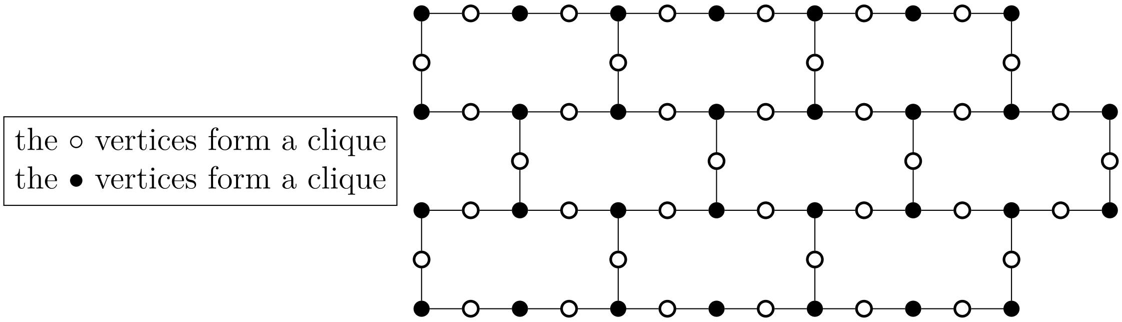


Figure 5.12 The graph $\overline{G_n}$ from the proof of Lemma 64 ($n = 3$ shown).

If X is a subset of either V_1 or V_2, say V_1, then every vertex of V_2 lies outside X. Since every vertex of V_1 has a neighbor in V_2, it follows that $\overline{G_n} \setminus X$ is connected in this case. Therefore X must contain at least one vertex of V_1 and at least one vertex of V_2. In $\overline{G_n}$, every vertex in V_1 has at most three neighbors in V_2 and vice versa, so X contains at most three vertices from V_1 and at most three vertices from V_2. In $\overline{G_n}$, every vertex in V_1 has at most three neighbors in V_2, and every vertex in V_2 has at least one neighbor in V_1, and $|V_2| > 12 = 9 + 3$. Hence, there must be a vertex in $V_2 \setminus X$ with a neighbor in $V_1 \setminus X$. Since V_1 and V_2 are cliques, it follows that $\overline{G_n} \setminus X$ is connected. This completes the proof. □

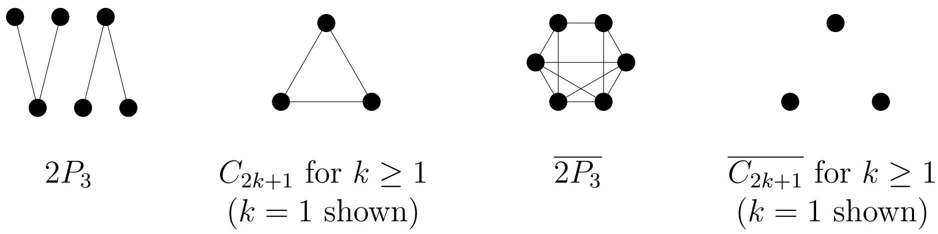


Figure 5.13 The forbidden induced subgraphs for the classes of $2P_3$-free bipartite graphs and $\overline{2P_3}$-free co-bipartite graphs mentioned in Lemma 65.

Lemma 65. *The class of $2P_3$-free bipartite atoms and the class of $\overline{2P_3}$-free co-bipartite atoms have unbounded clique-width (see Figure 5.13 for illustrations of the forbidden induced subgraphs).*

Proof. We adapt the construction of the graph G_n, which was used by Lozin and Volz [149] to show that the class of $2P_3$-free bipartite graphs has unbounded clique-width. For $n \geq 3$, construct the graph G_n as follows. Let the vertex set of G_n be $\{v_{i,j} \mid i \in \{0, \ldots, n\}, j \in \{1, \ldots, n\}\} \cup \{w_{i,j} \mid i \in \{1, \ldots, n\}, j \in \{0, \ldots, n\}\}$. For $i, j, k \in \{1, \ldots, n\}$, add an edge between $v_{i,j}$ and $w_{k,0}$ if $k \geq i$ and add an edge between $w_{i,j}$ and $v_{0,k}$ if $k \geq j$. For each $i, j \in \{1, \ldots, n\}$, add an edge between $v_{i,j}$ and $w_{i,j}$ and an edge between $v_{0,j}$ and $w_{i,0}$. Let G_n be the resulting graph. Then G_n is a $2P_3$-free bipartite graph and the family of such graphs has unbounded clique-width [149] (the former can also be seen by inspection and the latter follows from Lemma 56). Therefore the class of $2P_3$-free bipartite graphs has unbounded clique-width, and so Fact 2 implies that the class of $\overline{2P_3}$-free co-bipartite graphs has unbounded clique-width.

We observe that every graph in $\{\overline{2P_3}, \overline{C_3}, \overline{C_5}, \overline{C_7}, \ldots\}$ has no dominating vertex and no two non-adjacent vertices that are complete to the remainder of the graph. Therefore, by Lemma 60, it follows that the class of $\overline{2P_3}$-free co-bipartite atoms has unbounded clique-width. Now let H_n be the graph obtained from G_n by deleting $v_{n,n}$ and $w_{n,n}$ (see Figure 5.14) and note that H_n is a $2P_3$-free bipartite graph. By Fact 1, the family of graphs H_n has unbounded clique-width.

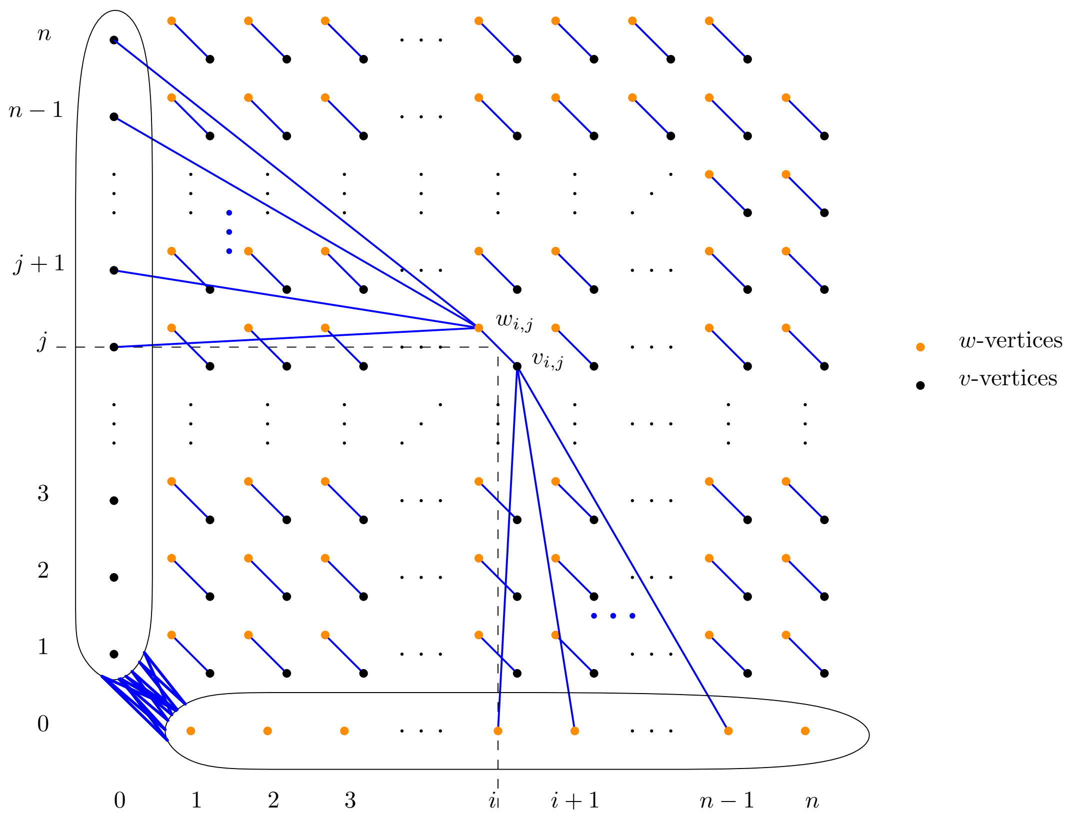


Figure 5.14 The graph H_n from the proof of Lemma 65. For clarity, the edges incident to $v_{i,j}$ and $w_{i,j}$ when $i, j \in \{1, \ldots, n\}$ are depicted for only one such pair of vertices.

We now prove that the class of $2P_3$-free bipartite atoms has unbounded clique-width. Let V be the set of vertices in the $2P_3$-free bipartite graph H_n of the form $v_{i,j}$ and let W be the set of vertices in H_n of the form $w_{i,j}$ and note that V and W are independent sets. Suppose, for contradiction, that H_n has a clique cut-set X. Since H_n is bipartite, every clique cut-set in H_n contains at most one vertex from each part, so $|X| \leq 2$. If X does not contain $v_{0,n}$, then every vertex in $W \setminus X$ is in the same component of $H_n \setminus X$. Since every vertex in V has at least two neighbors in W, and at most one vertex of W is in X, it follows that every vertex of $V \setminus X$ is in the same component of $H_n \setminus X$, and so $H_n \setminus X$ is connected. This contradiction implies that $v_{0,n} \in X$. By symmetry, $w_{n,0} \in X$. By construction, deleting $v_{0,n}$ and $w_{n,0}$ does not disconnect H_n, so H_n is indeed an atom. This completes the proof. □

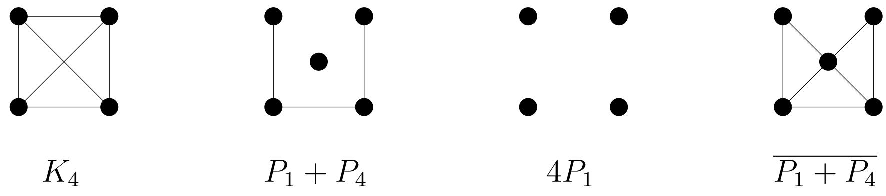


Figure 5.15 The forbidden induced subgraphs for the classes of $(K_4, P_1 + P_4)$-free graphs and $(4P_1, \overline{P_1 + P_4})$-free graphs mentioned in Lemma 66.

Lemma 66. *The class of $(K_4, P_1 + P_4)$-free atoms and the class of $(4P_1, \overline{P_1 + P_4})$-free atoms have unbounded clique-width (see Figure 5.15 for illustrations of the forbidden induced subgraphs).*

Proof. For this proof we use a construction that is implicit in the proof in [137, Theorem 3] that Graph Isomorphism is GI-complete on the class of $(K_4, P_1 + P_4)$-free graphs; we give an explicit construction. Consider a 1-subdivided wall of height $n \geq 2$. This graph is bipartite; let P and Q be the two parts of its bipartition with the vertices in Q being the vertices added by the subdivision. Consider a 3-subdivision of this 1-subdivided wall (so the resulting graph is a 7-subdivided wall). Partition the vertices introduced by this 3-subdivision as follows: let A be the set of vertices that are adjacent to vertices of P, let C be the set of vertices that are adjacent to vertices of Q, and let B be the set of remaining vertices introduced by the 3-subdivision (which have a neighbor in both A and C. Apply complementations to $P \cup C$, $Q \cup A$, and B (these sets will become cliques). Let H_n be the resulting graph (see also Figure 5.16) and note that $\overline{H_n}$ is $(K_4, P_1 + P_4)$-free and that the family of such graphs has unbounded clique-width [165] (the former statement can be seen by inspection and the latter can be seen by combining Lemma 57 and Fact 2). Therefore the class of $(K_4, P_1 + P_4)$-free graphs has unbounded clique-width. We observe that neither K_4 nor $P_1 + P_4$ contains a pair of false twins. Therefore, by Lemma 59, the class of $(K_4, P_1 + P_4)$-free atoms has unbounded clique-width.

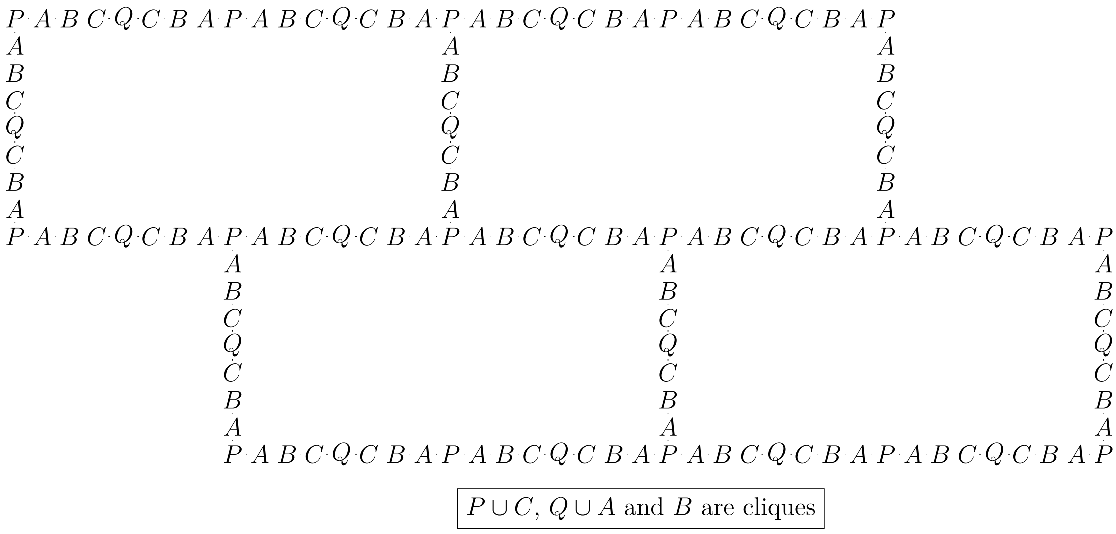


Figure 5.16 The graph H_n from the proof of Lemma 66 ($n = 2$ shown).

We now prove that the class of $(4P_1, \overline{P_1 + P_4})$-free atoms has unbounded clique-width. As $\overline{H_n}$ is $(K_4, P_1 + P_4)$-free, it follows that H_n is $(4P_1, \overline{P_1 + P_4})$-free. By Fact 2, it follows that the family of graphs H_n also has unbounded clique-width. It remains to show that H_n is an atom. Suppose, for contradiction, that H_n has a clique cut-set X. Recall that $P \cup C$, $Q \cup A$, and B are cliques. As A and C are anti-complete to each other, it follows that X contains vertices from at most one of A or C. Similarly, since P, Q and B are pairwise anti-complete, it follows that X contains vertices from at most one of P, Q or B. Note that every vertex from $P \cup Q \cup B$ has a neighbor in A and in C. Since A and C are cliques and X contains vertices in at most one of these sets, it follows that the vertices in $(P \cup Q \cup B) \setminus X$ all lie in the same component of $H_n \setminus X$. Similarly, every vertex of $A \cup C$ has a neighbor in both P and Q. Since X contains vertices from at most one of P or Q, it follows that the vertices in $(A \cup C) \setminus X$ are in the same component of $H_n \setminus X$ as the vertices of $(P \cup Q \cup B) \setminus X$. Therefore $H_n \setminus X$ is connected. This contradiction implies that H_n is indeed an atom. This completes the proof. ☐

$4P_1$ $\quad$ $\overline{3P_1 + P_2}$ $\quad$ $\overline{4P_1}$ $\quad$ $3P_1 + P_2$

Figure 5.17 The forbidden induced subgraphs for the classes of $(4P_1, \overline{3P_1 + P_2})$-free graphs and $(\overline{4P_1}, 3P_1 + P_2)$-free graphs mentioned in Lemma 67.

Lemma 67. *The class of $(4P_1, \overline{3P_1 + P_2})$-free atoms and the class of $(\overline{4P_1}, 3P_1 + P_2)$-free atoms have unbounded clique-width (see Figure 5.17 for illustrations of the forbidden induced subgraphs).*

Proof. We use the construction of [69] for proving that the class of $(4P_1, \overline{3P_1 + P_2})$-free graphs has unbounded clique-width. Let $n \geq 7$ and consider an $n \times n$ grid H_n and for $i, j \in \{0, \dots, n-1\}$, let $v_{i,j}$ be the vertex of H_n with x-coordinate i and y-coordinate j. For $k \in \{0, 1, 2\}$, let $V_k = \{v_{i,j} \mid i + j \equiv k \bmod 3\}$ (see also Figure 5.18 for a depiction of this 3-coloring). Apply a complementation to each V_k. Let G_n be the resulting graph. The resulting graph G_n is $(4P_1, \overline{3P_1 + P_2})$-free and the family of graphs G_n has unbounded clique-width [69] (the first of these statements can also be seen by inspection and the latter follows from combining Lemma 56 and Fact 2. By Fact 2, it follows that the family of graphs $\overline{G_n}$ also has unbounded clique-width.

It remains to show that G_n and $\overline{G_n}$ are atoms. Suppose, for contradiction, that G_n has a clique cut-set X. If $X \subseteq V_i$ for some $i \in \{0, 1, 2\}$, then all vertices of $G_n \setminus V_i$ are in the same component of $G_n \setminus X$. Since every vertex in V_i has at least one neighbor outside of V_i, it follows that every vertex of $G_n \setminus X$ is in the same component of $G_n \setminus X$ in this case, a contradiction. We may therefore assume that X contains vertices in at least two sets V_i. By construction, each vertex in a set V_i has at most two neighbors in each V_j for $j \in \{0, 1, 2\} \setminus \{i\}$. Therefore X has at most two vertices in each V_i. Since $n \geq 7$, there must be at least 15 vertices in V_0 that have neighbors in both V_1 and V_2 (see also

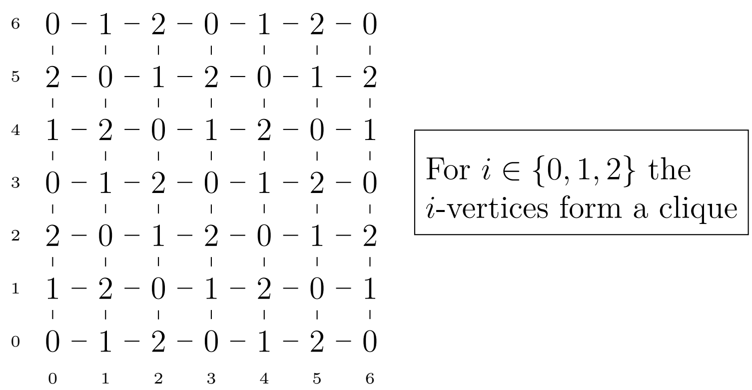


Figure 5.18 The graph G_n from the proof of Lemma 67 ($n = 7$ shown).

Figure 5.18). Since very vertex in $V_1 \cup V_2$ has at most two neighbors in V_0, there must be a vertex in $V_0 \setminus X$ that has a neighbor in both $V_1 \setminus X$ and $V_2 \setminus X$. Since each set V_i is a clique, it follows that $G_n \setminus X$ is connected. This contradiction implies that G_n is indeed an atom. Now suppose, for contradiction, that $\overline{G_n}$ has a clique cut-set X. Since V_0, V_1 and V_2 are independent sets in $\overline{G_n}$, X contains at most one vertex of any V_i. Since every vertex of V_i has at most two non-neighbors in each V_j for $j \in \{0, 1, 2\} \setminus \{i\}$, it follows that $\overline{G_n} \setminus X$ must be connected. This contradiction implies that $\overline{G_n}$ is indeed an atom. □

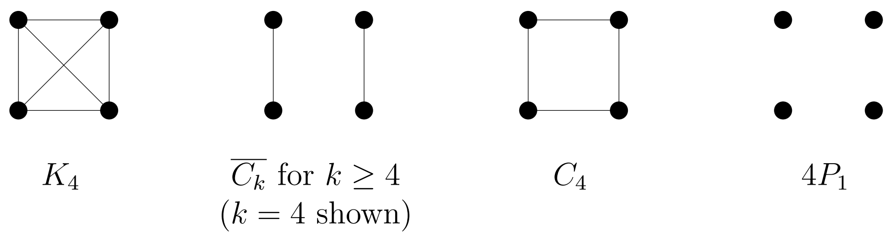


Figure 5.19 The forbidden induced subgraphs for the classes of K_4-free co-chordal graphs and $(C_4, 4P_1)$-free graphs mentioned in Lemma 68.

Lemma 68. *The class of K_4-free co-chordal atoms and the class of $(C_4, 4P_1)$-free atoms have unbounded clique-width (see Figure 5.19 for illustrations of the forbidden induced subgraphs).*

Proof. In [31, Theorem 11], Brandstädt et al. constructed a family of graphs G_n that are K_4-free co-chordal and have unbounded clique-width. The construction of G_n for $n \geq 3$ is as follows. Let the vertex set of G_n be $\{v_{i,j} \mid i, j \in \{0, \ldots, n\}, (i, j) \neq (0, 0)\}$. For $i, j, k \in \{1, \ldots, n\}$, add an edge between $v_{i,j}$ and $v_{k,0}$ if $k \geq i$ and add an edge between $v_{i,j}$ and $v_{0,k}$ if $k \geq j$. For each $i, j \in \{1, \ldots, n\}$, add an edge between $v_{i,0}$ and $v_{0,j}$. As shown in the proof of [31, Theorem 11], G_n is a K_4-free co-chordal graph and the family of such graphs has unbounded clique-width (the former property can also be seen by inspection and the latter follows from Lemma 56). Therefore the class of K_4-free co-chordal graphs had unbounded clique-width. We observe that neither K_4 nor $\overline{C_r}$ for any $r \geq 4$ contains a pair of false twins. Therefore, by Lemma 59, the class of K_4-free co-chordal atoms has unbounded clique-width.

We now prove that the class of $(C_4, 4P_1)$-free atoms has unbounded clique-width. Observe that the family of graphs of the form $\overline{G_n}$ is $4P_1$-free and chordal, and by Fact 2, it has unbounded clique-width. Now $\overline{G_n}$ is not an atom, since the set of vertices $\{v_{i,j} \mid i, j \in \{1, \ldots, n\}\}$ is a clique cut-set in $\overline{G_n}$. We construct a graph J_n from $\overline{G_{n+1}}$ as follows (see also Figure 5.20). Delete the vertices in the set $\{v_{1,i}, v_{i,1} \mid i \in \{1, \ldots, n+1\}$ and add the edge $v_{0,n+1}v_{n+1,0}$. Let J_n be the resulting graph. Now J_n contains $\overline{G_{n-1}}$ as an induced subgraph, so the family of graphs of the form J_n has unbounded clique-width. We claim that J_n is $(C_4, 4P_1)$-free. Note that $J_n \setminus \{v_{0,n+1}\}$ and $J_n \setminus \{v_{n+1,0}\}$ are induced subgraphs of $\overline{G_{n+1}}$, which is $(C_4, 4P_1)$-free. Therefore we only need to verify that there is no induced $4P_1$ or C_4 in J_n that contains both $v_{0,n+1}$ and $v_{n+1,0}$. Since $v_{0,n+1}$ is adjacent to $v_{n+1,0}$, there cannot be an induced $4P_1$ in J_n that contains both these vertices. Now $N(v_{0,n+1}) = \{v_{n+1,0}\} \cup \{v_{0,1}, \ldots, v_{0,n}\}$ and $N(v_{n+1,0}) = \{v_{0,n+1}\} \cup \{v_{1,0}, \ldots, v_{n,0}\}$. Since no vertex in $\{v_{0,1}, \ldots, v_{0,n}\}$ has a neighbor in $\{v_{1,0}, \ldots, v_{n,0}\}$ in J_n, it follows that J_n is indeed C_4-free.

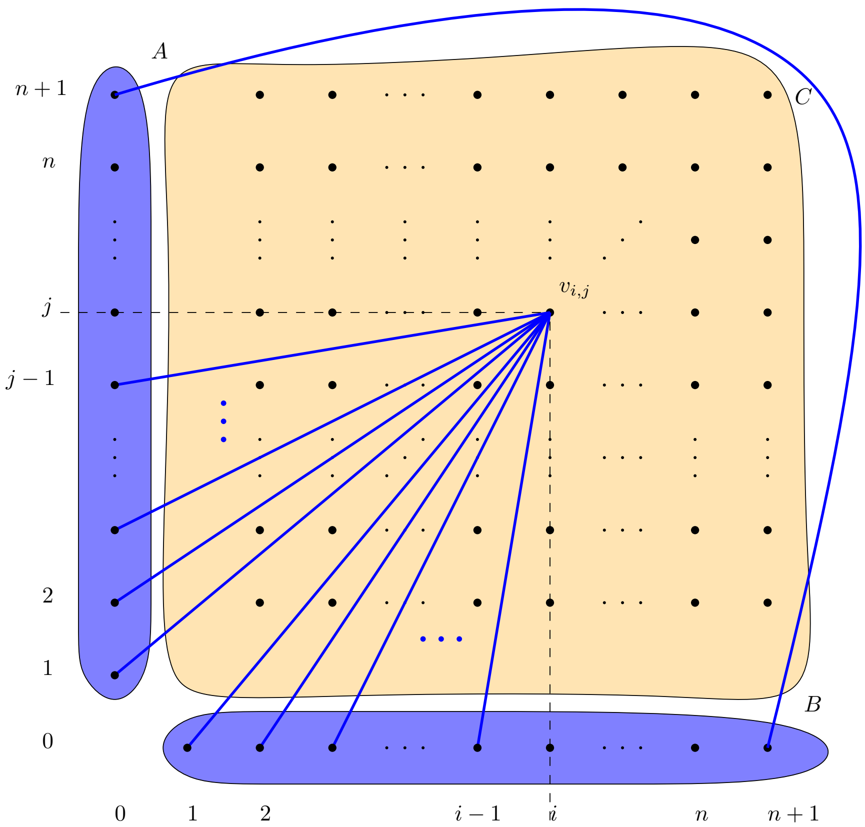


Figure 5.20 The graph J_n from the proof of Lemma 68. The sets A, B and C are cliques. For clarity, the edges between $A \cup B$ and C are depicted for only one vertex in C.

It remains to show that J_n is an atom. Suppose, for contradiction, that X is a clique cut-set of J_n. First, suppose that $v_{0,n+1}$ is in X. Then $v_{1,0} \notin X$ and $v_{n+1,n+1} \notin X$ as $v_{0,n+1}$ is non-adjacent to these vertices. As $v_{1,0}$ and $v_{n+1,n+1}$ are adjacent and every vertex in $J_n \setminus \{v_{0,n+1}\}$ is adjacent to at least one of these vertices, we find that X is not a clique cut-set. We may therefore assume that $v_{0,n+1} \notin X$ and by symmetry, that $v_{n+1,0} \notin X$. We partition the vertices $v_{i,j}$ in J_n into three sets A, B and C, if $j = 0$, $i = 0$ or $i, j \neq 0$, respectively, and note that each of these sets is a clique. Since A and B are cliques and the vertices $v_{0,n+1}$ and $v_{n+1,0}$ are adjacent, it follows that all vertices in $(A \cup B) \setminus X$ are in the same component of $J_n \setminus X$. Note that every vertex from C has at least one neighbor in both A and B. However, X cannot contain vertices from both A and B since $A \setminus \{v_{0,n+1}\}$ and $B \setminus \{v_{n+1,0}\}$ are anti-complete. Therefore $J_n \setminus X$ is connected. This contradiction implies that J_n is an atom. $\square$

$\overline{P_1 + P_4}$ $\quad$ $P_1 + 2P_2$ $\quad$ $P_1 + P_4$ $\quad$ $\overline{P_1 + 2P_2}$

Figure 5.21 The forbidden induced subgraphs for the classes of $(\overline{P_1 + P_4}, P_1 + 2P_2)$-free graphs and $(P_1 + P_4, \overline{P_1 + 2P_2})$-free graphs mentioned in Lemma 69.

Lemma 69. *The class of* $(\overline{P_1 + P_4}, P_1 + 2P_2)$*-free atoms and the class of* $(P_1 + P_4, \overline{P_1 + 2P_2})$*-free atoms have unbounded clique-width (see Figure 5.21 for illustrations of the forbidden induced subgraphs).*

Proof. We use the construction from [17], which was used to show that $(\overline{P_1 + P_4}, P_1 + 2P_2)$-free graphs have unbounded clique-width. We copy this construction below. Let $t \geq 2$ and let G be the $t \times t$ square grid. Let $v_1^G, \ldots, v_n^G$ be the vertices of G and let $e_1^G, \ldots, e_m^G$ be the edges of G. We construct a graph $q(G)$ from G as follows (see also Figure 5.22):

1. Create a complete multi-partite graph with partition $(A_1^G, \ldots, A_n^G)$, where $|A_i^G| = d_G(v_i^G)$ for $i \in \{1, \ldots, n\}$ and let $A^G = \bigcup A_i^G$.

2. Create a complete multi-partite graph with partition $(B_1^G, \ldots, B_m^G)$, where $|B_i^G| = 2$ for $i \in \{1, \ldots, m\}$ and let $B^G = \bigcup B_i^G$.

3. Take the disjoint union of the two graphs above, then for each edge $e_i^G = v_{i_1}^G v_{i_2}^G$ in G in turn, add an edge from one vertex of B_i^G to a vertex of $A_{i_1}^G$ and an edge from the other vertex of B_i^G to a vertex of $A_{i_2}^G$. Do this in such a way that the edges added between A^G and B^G form a perfect matching.

In [17] it was shown that the graph $q(G)$ is $(\overline{P_1 + P_4}, P_1 + 2P_2)$-free and that the clique-width of such graphs is unbounded. Therefore the class of $(\overline{P_1 + P_4}, P_1 + 2P_2)$-free graphs has unbounded clique-width. We observe that neither $\overline{P_1 + P_4}$ nor $P_1 + 2P_2$ contains a pair of false twins. Therefore, by Lemma 59, it follows that the class of $(\overline{P_1 + P_4}, P_1 + 2P_2)$-free atoms has unbounded clique-width.

We now prove that the class of $(P_1 + P_4, \overline{P_1 + 2P_2})$-free atoms has unbounded clique-width. By Fact 2, the class of $(P_1 + P_4, \overline{P_1 + 2P_2})$-free graphs of the form $\overline{q(G)}$ also has unbounded clique-width. It therefore suffices to show that $\overline{q(G)}$ is an atom. Suppose, for

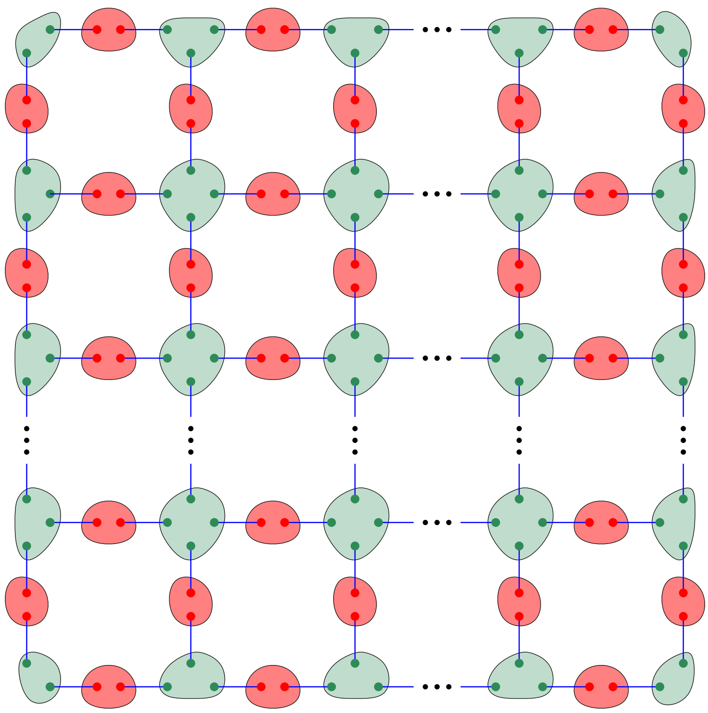


Figure 5.22 The graph $q(G)$ in Lemma 69. Edges between partition classes A_i^G and edges between partition classes B_i^G are not shown.

contradiction, that X is a clique cut-set in $\overline{q(G)}$. In $\overline{q(G)}$, A^G induces a disjoint union of cliques of the form A_i^G, and B^G induces a disjoint union of cliques of the form B_i^G. Therefore $X \in A_i^G \cup B_j^G$ for some i, j. Let $k \neq i$ and $\ell \neq j$. Then in $\overline{q(G)}$ every vertex in B^G has a neighbor in A_k^G and every vertex in A^G has a neighbor in B_ℓ^G. Since A_k^G and B_ℓ^G are cliques, it follows that $\overline{q(G)} \setminus X$ is connected, a contradiction. Therefore $\overline{q(G)}$ is indeed an atom. □

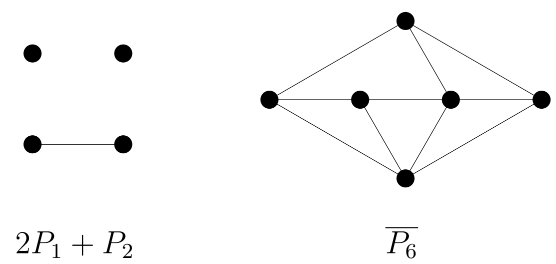


Figure 5.23 The forbidden induced subgraphs for the class of $(2P_1 + P_2, \overline{P_6})$-free graphs mentioned in Lemma 70.

Lemma 70. *The class of $(2P_1 + P_2, \overline{P_6})$-free atoms has unbounded clique-width (see Figure 5.23 for illustrations of the forbidden induced subgraphs).*

Proof. By Theorem 55.2(iv), the class of $(2P_1 + P_2, \overline{P_6})$-free graphs has unbounded clique-width. We observe that neither $2P_1 + P_2$ nor $\overline{P_6}$ has a dominating vertex or a pair of non-adjacent vertices that are complete to the remainder of the graph. Therefore, by Lemma 60, it follows that the class of $(2P_1 + P_2, \overline{P_6})$-free atoms has unbounded clique-width. $\square$

$\overline{P_1 + P_4}$ $\quad$ P_6

Figure 5.24 The forbidden induced subgraphs for the class of $(\overline{P_1 + P_4}, P_6)$-free graphs mentioned in Lemma 71.

Lemma 71. *The class of $(\overline{P_1 + P_4}, P_6)$-free atoms has unbounded clique-width (see Figure 5.24 for illustrations of the forbidden induced subgraphs).*

Proof. By Theorem 55.2(iv), the class of $(\overline{P_1 + P_4}, P_6)$-free graphs has unbounded clique-width. We observe that neither $\overline{P_1 + P_4}$ nor P_6 contains a pair of false twins. Therefore, by Lemma 59, it follows that the class of $(\overline{P_1 + P_4}, P_6)$-free atoms has unbounded clique-width. $\square$

$\overline{P_2 + P_3}$ $\quad$ $\overline{P_1 + 2P_2}$ $\quad$ $P_1 + 2P_2$ $\quad$ P_6

Figure 5.25 The forbidden induced subgraphs for the class of $(\overline{P_2 + P_3}, \overline{P_1 + 2P_2}, P_1{+}2P_2, P_6)$-free graphs mentioned in Lemma 72.

Lemma 72. *The class of $(\overline{P_2 + P_3}, \overline{P_1 + 2P_2}, P_1 + 2P_2, P_6)$-free atoms has unbounded clique-width (see Figure 5.25 for illustrations of the forbidden induced subgraphs).*

Proof. Consider a 1-subdivision of a wall of height $n \geq 2$. Let A be the set of original vertices of the wall and let B be the set of vertices introduced by the subdivision. Apply a complementation to A and add a vertex x complete to B. Let H_n be the resulting graph (see Figure 5.26). By Lemma 57, combined with Facts 1 and 2, the family of such graphs has unbounded clique-width. Note that x is complete to the independent set B and anti-complete to the clique A. Every vertex in B has exactly two neighbors in A and every vertex in A has either two or three neighbors in B. Furthermore, no two vertices in B have the same pair of neighbors in A.

We prove that H_n is an atom. Suppose, for contradiction, that X is a clique cut-set of H_n. If $x \in X$, then X can contain at most one other vertex (which must be in the independent set B). Since every vertex in B has neighbors in the clique A, it follows that

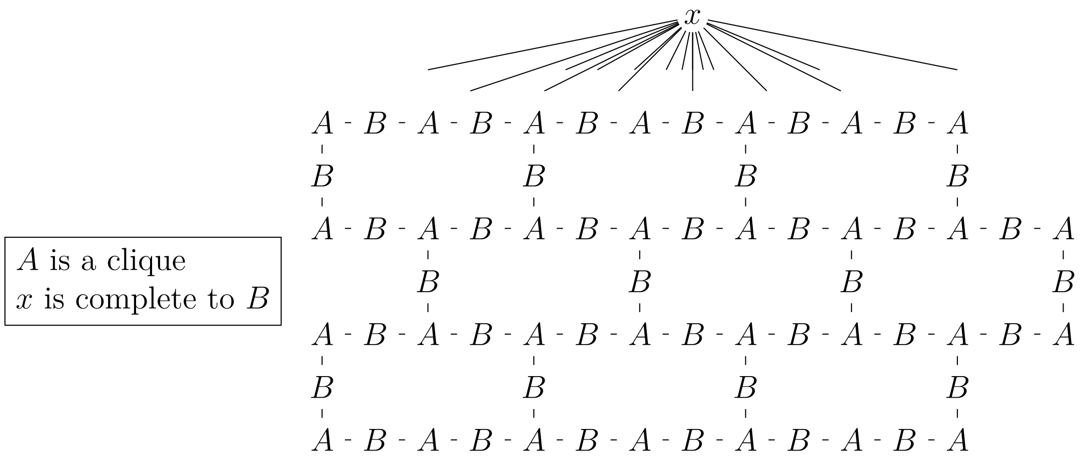


Figure 5.26 The graph H_n from the proof of Lemma 72 ($n = 3$ shown). Vertices are denoted A or B if they are in the corresponding set.

$H_n \setminus X$ is connected. We may therefore assume that $x \notin X$. Since B is an independent set, $|B \cap X| \leq 1$, so every vertex of A has a neighbor in $B \setminus X$. Since x is complete to B, it follows that every vertex of $B \setminus X$ is in the same component of $B \setminus X$ as x, and so $H_n \setminus X$ is connected, a contradiction. We conclude that H_n is indeed an atom.

It remains to show that H_n is $(\overline{P_2+P_3}, \overline{P_1+2P_2}, P_1{+}2P_2, P_6)$-free. Note that $H_n[A \cup B]$ is a split graph, so it is $(2P_2, \overline{2P_2})$-free. Therefore every induced $2P_2$ or $\overline{2P_2}$ in H_n contains the vertex x. Suppose, for contradiction, that H_n contains an induced $\overline{P_2+P_3}$ or $\overline{P_1+2P_2}$, say on vertex set Y. Since $\overline{P_2+P_3}$ and $\overline{P_1+2P_2}$ each contain an induced $\overline{2P_2}$, it follows that $x \in Y$. Since x has two neighbors and one non-neighbor in this $\overline{2P_2}$, this $\overline{2P_2}$ consists of the vertex x, two vertices in B and one vertex in A. Now Y contains one more vertex y, which is adjacent to either three or four of the remaining vertices of Y. Now y cannot be in B, since B is an independent set and there are two vertices in $(B \cap Y) \setminus \{y\}$, so $y \in A$. Therefore $A \cap Y$ contains two vertices with two common neighbors in B, contradicting the fact that no two vertices of B have the same two neighbors in A. We conclude that H_n is indeed $(\overline{P_2+P_3}, \overline{P_1+2P_2})$-free. Now suppose, for contradiction, that H_n contains an induced P_1+2P_2 or an induced P_6, say on vertex set Y. Since P_1+2P_2 and P_6 each contain an induced $2P_2$, we find that $x \in Y$. Every vertex in P_1+2P_2 and P_6 has at least three non-neighbors. Since x is complete to B, it follows that $|A \cap Y| \geq 3$. But A is a clique and so $H_n[A \cap Y]$ contains a K_3, which is a contradiction, since $P_1{+}2P_2$ and P_6 are K_3-free. We conclude that H_n is $(P_1{+}2P_2, P_6)$-free. Hence, H_n is $(\overline{P_2+P_3}, \overline{P_1+2P_2}, P_1+2P_2, P_6)$-free. □

Lemma 73. *The class of $(2P_2, \overline{P_2+P_4})$-free atoms has unbounded clique-width (see Figure 5.27 for illustrations of the forbidden induced subgraphs).*

Proof. Let $n \geq 2$ and construct the graph G_n as follows (see also Figure 5.28). Let the vertex set of G_n be $\{v_{i,j} \mid i,j \in \{0,\ldots,n\}, (i,j) \neq (0,0)\} \cup \{v_{0,n+1}\}$. For $i,j,k \in \{1,\ldots,n\}$, add an edge between $v_{i,j}$ and $v_{k,0}$ if $k \geq i$, add an edge between $v_{i,j}$ and $v_{0,k}$ if $k \geq j$, and add an edge between $v_{i,j}$ and $v_{0,n+1}$. Let $A = \{v_{i,0} \mid i \in \{1,\ldots,n\}\}$, $B = \{v_{0,j} \mid j \in \{1,\ldots,n+1\}\}$, and $C = \{v_{i,j} \mid i,j \in \{1,\ldots,n\}\}$. Apply a bipartite

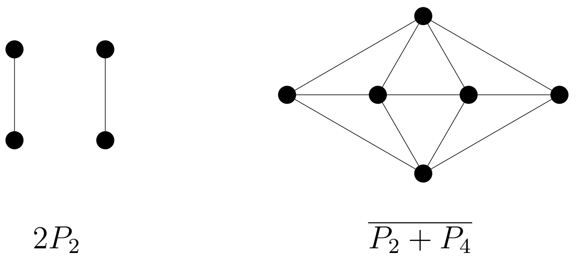


Figure 5.27 The forbidden induced subgraphs for the class of $(2P_2, \overline{P_2 + P_4})$-free graphs mentioned in Lemma 73.

complementation between A and B and apply a complementation to A. By Lemma 56 combined with Facts 1, 2 and 3, the family of graphs G_n has unbounded clique-width.

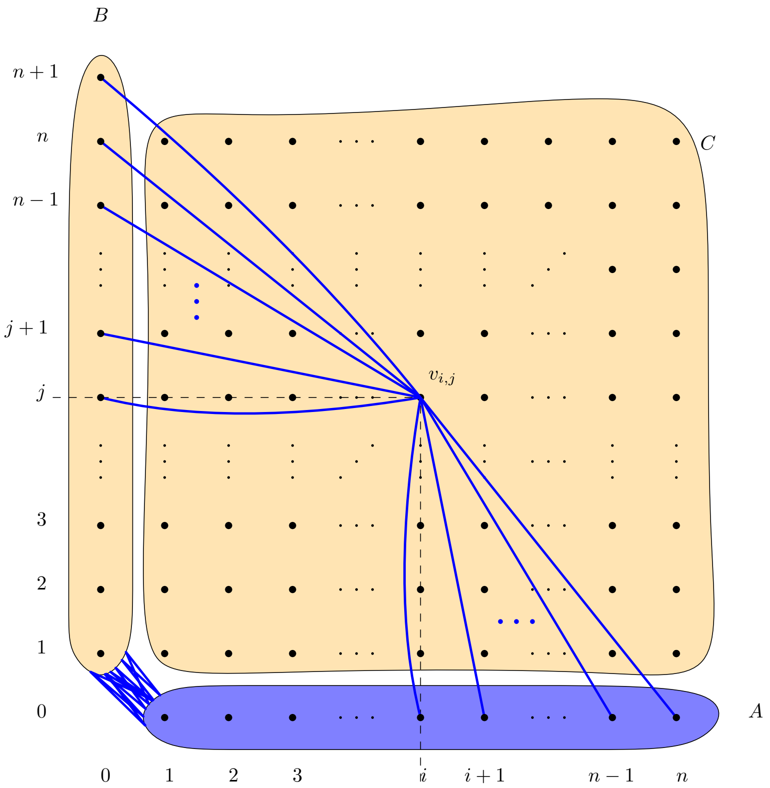
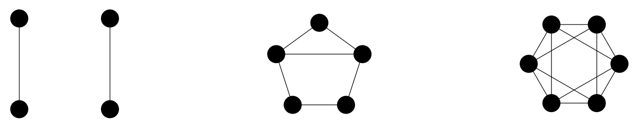


Figure 5.28 The graph G_n constructed in the proof of Lemma 73. The set A is a clique. For clarity, the edges between $A \cup B$ and C are depicted for only one vertex in C.

We claim that G_n is an atom. Suppose, for contradiction, that X is a clique cut-set of G_n. Since $v_{0,n}$ and $v_{0,n+1}$ are non-adjacent, but complete to C, it follows that every vertex of C is in the same component of $G_n \setminus X$. Since C is independent, at most one vertex of C is in X. Since every vertex in $A \cup B$ has at least two neighbors in C, it follows that every vertex of $G_n \setminus X$ is in the same component of $G_n \setminus X$, a contradiction. Therefore G_n is indeed an atom.

Now suppose, for contradiction, that G_n contains an induced subgraph isomorphic to $2P_2$, say on vertex set Y. Since $G_n[A \cup C]$ is a split graph, it is $2P_2$-free, so Y must contain at least one vertex of B. Since $G_n[B \cup C]$ is a bipartite chain graph, it is $2P_2$-free, so Y contains at least one vertex of A. Since A is complete to B, it follows that Y contains exactly one vertex of A and exactly one vertex of B, and these two vertices are adjacent. Therefore $Y \cap C$ contains two adjacent vertices, contradicting the fact that C is independent. We conclude that G_n is $2P_2$-free.

Next suppose, for contradiction, that G_n contains an induced subgraph isomorphic to $\overline{P_2 + P_4}$, say on vertex set Y. Since $\overline{P_2 + P_4}$ is $3P_1$-free, and B and C are independent, it follows that $|B \cap Y| \leq 2$ and $|C \cap Y| \leq 2$. Therefore $|A \cap Y| \geq 6 - 2 - 2 = 2$. Now $G_n[A \cup B]$ and $G_n[A \cup C]$ are split graphs and therefore C_4-free. Since $\overline{P_2 + P_4}$ contains an induced $\overline{2P_2} = C_4$, it follows that $|C \cap Y| \geq 1$ and $|B \cap Y| \geq 1$. Since A is a clique that is complete to B and $\overline{P_2 + P_4}$ is K_4-free, it follows that $|A \cap Y| \leq 2$. Therefore $|A \cap Y| = |B \cap Y| = |C \cap Y| = 2$. Since B and C are cliques in $\overline{G_n}$, we observe that all vertices in $B \cap Y$ are in the same component of $\overline{G_n}[Y]$, and all vertices in $C \cap Y$ are in the same component of $\overline{G_n}[Y]$. Furthermore, since in $\overline{G_n}$ the set A is independent and anti-complete to B, the vertices in $(A \cup C) \cap Y$ form the P_4-component of $\overline{G_n}[Y]$, and vertices in $B \cap Y$ form the P_2-component of $\overline{G_n}[Y]$. Therefore $\overline{G_n}[(A \cup C) \cap Y]$ is isomorphic to P_4, which means that in $\overline{G_n}$ there must be two vertices in the clique C that have private neighbors in the independent set A. By the construction of $\overline{G_n}$, the vertices in C can be linearly ordered according to their neighborhood in A, a contradiction. Therefore G_n is indeed $\overline{P_2 + P_4}$-free. □



$2P_2$ $\quad$ $\overline{P_5}$ $\quad$ $\overline{3P_2}$

Figure 5.29 The forbidden induced subgraphs for the class of $(2P_2, \overline{P_5}, \overline{3P_2})$-free graphs mentioned in Lemma 74.

Lemma 74. *The class of $(2P_2, \overline{P_5}, \overline{3P_2})$-free atoms has unbounded clique-width (see Figure 5.29 for illustrations of the forbidden induced subgraphs).*

Proof. Consider a wall of height $k \geq 2$ and note that it is a bipartite graph, say with parts A and B. Apply a complementation to A and add two vertices x and y that are complete to $A \cup B$. Let H_k be the resulting graph (see Figure 5.30). By Lemma 57, combined with Facts 1 and 2, the class of such graphs has unbounded clique-width.

We claim that H_k is an atom. Suppose, for contradiction, that X is a clique cut-set of H_k. Since x is non-adjacent to y, at most one of x and y is in X. Without loss of generality, we may assume that $y \notin X$. Now y is adjacent to every vertex of $A \cup B$, so $H_k[\{y\} \cup (A \cup B) \setminus X]$ is connected. Since $A \cup B$ is not a clique, there must be at least one vertex in $(A \cup B) \setminus X$. Therefore, if $x \notin X$ then x is in the same component of $H_k \setminus X$ as y is. It follows that $H_k \setminus X$ is connected, a contradiction. Therefore H_k is indeed an atom.

It remains to show that H_k is $(2P_2, \overline{P_5}, \overline{3P_2})$-free. First, note that $H_k \setminus \{x\}$ and $H_k \setminus \{y\}$ are split graphs, so they are $(2P_2, \overline{2P_2})$-free, and therefore $(2P_2, \overline{P_5}, \overline{P_1 + 2P_2})$-free and

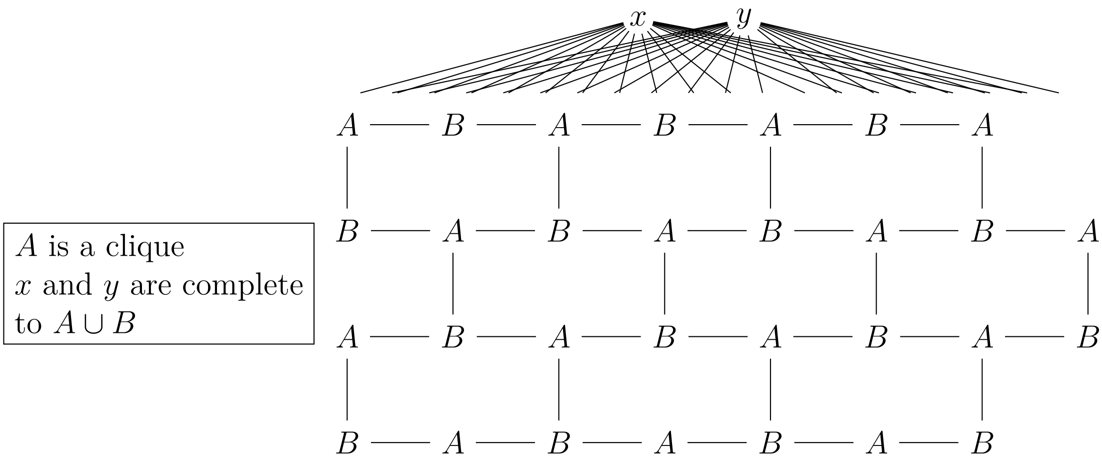


Figure 5.30 The graph H_k from the proof of Lemma 74 ($k = 3$ shown). Vertices are denoted A or B if they are in the corresponding set.

note that this also implies that H_k is $\overline{3P_2}$-free. Therefore, if H_k contains an induced $2P_2$ or $\overline{P_5}$, then this induced copy must contain both x and y. Since x and y are false twins in H_k, but $2P_2$ and $\overline{P_5}$ do not contain two vertices that are false twins, it follows that H_k is $(2P_2, \overline{P_5})$-free. This completes the proof. □

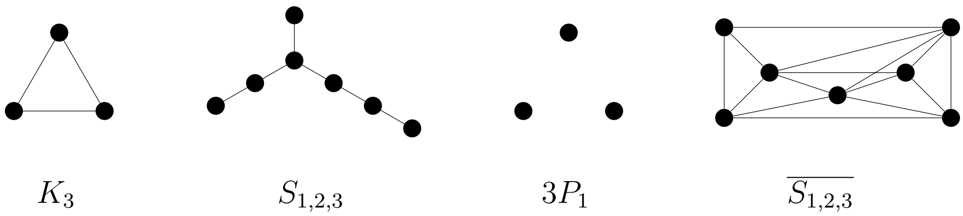


Figure 5.31 The forbidden induced subgraphs for the classes of $(K_3, S_{1,2,3})$-free graphs and $(3P_1, \overline{S_{1,2,3}})$-free graphs mentioned in Lemma 75.

It is not known whether the clique-width of $(K_3, S_{1,2,3})$-free graphs is bounded or unbounded. Recall that this case is equivalent to the open case for $(3P_1, \overline{S_{1,2,3}})$-free graphs; see also Open Problem 1. As a final result of this section, we observe that the boundedness for these cases also matches their atom counterparts.

Lemma 75. *The class of* $(K_3, S_{1,2,3})$*-free atoms has bounded clique-width if and only if the class of* $(K_3, S_{1,2,3})$*-free graphs has bounded clique-width if and only if the class of* $(3P_1, \overline{S_{1,2,3}})$*-free atoms has bounded clique-width (see Figure 5.31 for illustrations of the forbidden induced subgraphs).*

Proof. We first observe that neither K_3 nor $S_{1,2,3}$ contains a pair of false twins. Therefore, by Lemma 59, the class of $(K_3, S_{1,2,3})$-free atoms has bounded clique-width if and only if the class of $(K_3, S_{1,2,3})$-free graphs has bounded clique-width. By Fact 2, the class of $(K_3, S_{1,2,3})$-free graphs has bounded clique-width if and only if the class of $(3P_1, \overline{S_{1,2,3}})$-free graphs has bounded clique-width.

We observe that neither $3P_1$ nor $\overline{S_{1,2,3}}$ has a dominating vertex or a pair of non-adjacent vertices that are complete to the remainder of the graph. Therefore, we have that by Lemma 60, the class of $(3P_1, \overline{S_{1,2,3}})$-free graphs has bounded clique-width if and only if the class of $(3P_1, \overline{S_{1,2,3}})$-free atoms has bounded clique-width. □

5.6 The Proof of Theorem 7

Recall the definition of equivalent bigenic classes given at the start of Section 5.4. To make Theorem 7 easier to compare to Theorem 55, in this section we will use the following reformulation of it, where we group classes together if they are equivalent, and we will prove this reformulated version of the theorem instead (it is easy to verify that Theorems 7 and 76 cover the same graph classes).

Theorem 76. *Let $\mathcal{G}$ be a class of graphs defined by two forbidden induced subgraphs.*

1. *The class of atoms in $\mathcal{G}$ has bounded clique-width if it is equivalent to a class of (H_1, H_2)-free graphs such that one of the following holds:*

   (i) H_1 *or* $H_2 \subseteq_i P_4$

   (ii) $H_1 = K_s$ *and* $H_2 = tP_1$ *for some* $s, t \geq 1$

   (iii) $H_1 \subseteq_i \text{paw}$ *and* $H_2 \subseteq_i K_{1,3}+3P_1,\ K_{1,3}+P_2,\ P_1+P_2+P_3,\ P_1+P_5,\ P_1+S_{1,1,2},\ P_2+P_4,\ P_6,\ S_{1,1,3}$ *or* $S_{1,2,2}$

   (iv) $H_1 \subseteq_i \text{diamond}$ *and* $H_2 \subseteq_i P_1+2P_2,\ 3P_1+P_2$ *or* P_2+P_3

   (v) $H_1 \subseteq_i \text{gem}$ *and* $H_2 \subseteq_i P_1+P_4$ *or* P_5

   (vi) $H_1 \subseteq_i K_3+P_1$ *and* $H_2 \subseteq_i K_{1,3}$*, or*

   (vii) $H_1 \subseteq_i \overline{2P_1+P_3}$ *and* $H_2 \subseteq_i 2P_1+P_3$.

2. *The class of atoms in $\mathcal{G}$ has bounded clique-width if $\mathcal{G}$ is a subclass of the class of:*

   (i) $(P_6, \overline{2P_2})$*-free graphs or*

   (ii) $(2P_2, \overline{P_2+P_3})$*-free graphs.*

3. *The class of atoms in $\mathcal{G}$ has unbounded clique-width if it is equivalent to a class of (H_1, H_2)-free graphs such that one of the following holds:*

   (i) $H_1 \notin \mathcal{S}$ *and* $H_2 \notin \mathcal{S}$

   (ii) $H_1 \notin \overline{\mathcal{S}}$ *and* $H_2 \notin \overline{\mathcal{S}}$

   (iii) $H_1 \supseteq_i K_3+P_1$ *and* $H_2 \supseteq_i 4P_1$ *or* $2P_2$

   (iv) $H_1 \supseteq_i \text{diamond}$ *and* $H_2 \supseteq_i K_{1,3},\ 5P_1$ *or* P_2+P_4

   (v) $H_1 \supseteq_i K_3$ *and* $H_2 \supseteq_i 2P_1+2P_2,\ 2P_1+P_4,\ 4P_1+P_2,\ 3P_2$ *or* $2P_3$

   (vi) $H_1 \supseteq_i K_4$ *and* $H_2 \supseteq_i P_1+P_4,\ 3P_1+P_2$ *or* $2P_2$*, or*

   (vii) $H_1 \supseteq_i \text{gem}$ *and* $H_2 \supseteq_i P_1+2P_2$.

4. *The class of atoms in $\mathcal{G}$ has unbounded clique-width if it contains the class of (H_1, H_2)-free graphs such that one of the following holds:*

   (i) $H_1 \supseteq_i$ diamond *and* $H_2 \supseteq_i P_1 + P_6$

   (ii) $H_1 \supseteq_i 2P_1 + P_2$ *and* $H_2 \supseteq_i \overline{P_6}$

   (iii) $H_1 \supseteq_i$ gem *and* $H_2 \supseteq_i P_6$

   (iv) $H_1 \supseteq_i P_1 + 2P_2$ *or* P_6 *and* $H_2 \supseteq_i \overline{P_1 + 2P_2}$ *or* $\overline{P_2 + P_3}$*, or*

   (v) $H_1 \supseteq_i 2P_2$ *and* $H_2 \supseteq_i \overline{P_2 + P_4}$, $\overline{3P_2}$ *or* $\overline{P_5}$*.*

Proof. We start by considering the bounded cases. Theorem 76.1 follows immediately from Theorem 55.1. Theorem 76.2(i) follows from the fact that $(P_6, \overline{2P_2})$-free atoms have bounded clique-width [101]. Theorem 76.2(ii) follows from the fact that $(2P_2, \overline{P_2 + P_3})$-free atoms have bounded clique-width (Theorem 6). Next, we consider the unbounded cases. Theorem 76.3(i) and Theorem 76.3(ii) follow from Lemma 58. Theorem 76.3(iii) follows from Lemma 61. Theorem 76.3(iv) follows from Lemmas 61, 62 and 63. Theorem 76.3(v) follows from Lemmas 64 and 65. Theorem 76.3(vi) follows from Lemma 66, 67 and 68. Theorem 76.3(vii) follows from Lemma 69. Theorem 76.4(i) follows from Lemma 63. Theorem 76.4(ii) follows from Lemma 70. Theorem 76.4(iii) follows from Lemma 71. Theorem 76.4(iv) follows from Lemma 72. Theorem 76.4(v) follows from Lemmas 73 and 74. □

In the open problem below, the cases marked with a * are those for which even the boundedness of clique-width of the whole class of (H_1, H_2)-free graphs is unknown (see also Open Problem 1 in Section 5.4).

Open Problem 2. *Does the class of (H_1, H_2)-free atoms have bounded clique-width if*

(i) $H_1 =$ diamond *and* $H_2 = P_6$

(ii) $H_1 = C_4$ *and* $H_2 \in \{P_1 + 2P_2, P_2 + P_4, 3P_2\}$

(iii) $H_1 = \overline{P_1 + 2P_2}$ *and* $H_2 \in \{2P_2, P_2 + P_3, P_5\}$

(iv) $H_1 = \overline{P_2 + P_3}$ *and* $H_2 \in \{P_2 + P_3, P_5\}$

**(v)* $H_1 = K_3$ *and* $H_2 \in \{P_1 + S_{1,1,3}, S_{1,2,3}\}$

**(vi)* $H_1 = 3P_1$ *and* $H_2 = \overline{P_1 + S_{1,1,3}}$

**(vii)* $H_1 =$ diamond *and* $H_2 \in \{P_1 + P_2 + P_3, P_1 + P_5\}$

**(viii)* $H_1 = 2P_1 + P_2$ *and* $H_2 \in \{\overline{P_1 + P_2 + P_3}, \overline{P_1 + P_5}\}$

**(ix)* $H_1 =$ gem *and* $H_2 = P_2 + P_3$*, or*

**(x)* $H_1 = P_1 + P_4$ *and* $H_2 = \overline{P_2 + P_3}$*.*

Olariu [156] proved that every connected $\overline{P_1 + P_3}$-free graph is either K_3-free or complete multi-partite. Since complete multi-partite graphs and their complements have bounded clique-width, when looking at boundedness of clique-width of a hereditary class, forbidding $\overline{P_1 + P_3}$ as an induced subgraph is equivalent to forbidding K_3 and forbidding $P_1 + P_3$ is equivalent to forbidding $3P_1$. Thus, when studying boundedness of clique-width we may assume that we never explicitly forbid $\overline{P_1 + P_3}$ or P_1+P_3. Furthermore, by Lemma 75, the class of $(K_3, S_{1,2,3})$-free atoms has bounded clique-width if and only if the class of $(3P_1, \overline{S_{1,2,3}})$-free atoms has bounded clique-width, so we may assume $\{H_1, H_2\} \neq \{3P_1, \overline{S_{1,2,3}}\}$. We now state the following theorem.

Theorem 77. *Let H_1 and H_2 be graphs (which are not isomorphic to $\overline{P_1 + P_3}$ or $P_1 + P_3$) with $\{H_1, H_2\} \neq \{3P_1, \overline{S_{1,2,3}}\}$ and let $\mathcal{G}$ be the class of (H_1, H_2)-free graphs. Then (un)boundedness of clique-width for atoms in $\mathcal{G}$ does not follow from Theorem 76 if and only if this class is listed in Open Problem 2.*

Proof. First, note that Theorem 76 does not specify the (un)boundedness of clique-width for atoms in any of the classes listed in Open Problem 2.

Consider the classes listed in Open Problem 1. For all bigenic classes $\mathcal{G}$ for which the (un)boundedness of clique-width of general graphs is not listed in Theorem 55, an equivalent class is listed in Open Problem 1 (see [71] and [77]). Since the results in Theorem 76.2 do not solve these cases when restricted to atoms, these classes (and their complements, apart from the $H_1 = 3P_1$, $H_2 = \overline{S_{1,2,3}}$) appear in Open Problem 2.(v)-(x). The only other classes we need to consider are those for which Theorem 55.2 states that the class $\mathcal{G}$ has unbounded clique-width, but the class of atoms in $\mathcal{G}$ might not have unbounded clique-width.

There are two classes listed in Theorem 55.2 that are not listed in Theorem 76.3, namely the class of $(\overline{2P_2}, 2P_2)$-free graphs and the class of $(\overline{2P_1 + P_2}, P_6)$-free graphs. The class of $(\overline{2P_2}, 2P_2)$-free graphs is only equivalent to itself. The class of $(\overline{2P_1 + P_2}, P_6)$-free graphs equivalent to only one other class, namely the class of $(2P_1 + P_2, \overline{P_6})$-free graphs. However, the class of $(2P_1 + P_2, \overline{P_6})$-free atoms has unbounded clique-width by Theorem 76.4(ii). We therefore only need to consider the class of $(\overline{2P_2}, 2P_2)$-free graphs and the class of $(2P_1 + P_2, \overline{P_6})$-free, together with any bigenic classes $\mathcal{G}'$ that are extensions of these classes such that Theorem 76 does not specify that the atoms of $\mathcal{G}'$ have unbounded clique-width.

We start by considering extensions of the classes of $(\overline{2P_1 + P_2}, P_6)$-free graphs. Consider graphs H_1, H_2 with $\overline{2P_1 + P_2} \subseteq_i H_1$ and $P_6 \subseteq_i H_2$ such that the class of (H_1, H_2)-free atoms has bounded clique-width, but Theorem 76 does not state that (H_1, H_2)-free atoms have unbounded clique-width. By Theorem 76.3(ii), it follows that $\overline{H_1} \in \mathcal{S}$. By Theorem 76.3(iii), it follows that $\overline{H_1}$ is $K_{1,3}$-free, so it is a linear forest. By Theorem 76.3(vi), it follows that $\overline{H_1}$ is $4P_1$-free. By Theorem 76.4(iii), $\overline{H_1}$ must be $(P_1 + P_4)$-free. By Theorem 76.4(iv), $\overline{H_1}$ must be $(P_1 + 2P_2, P_2 + P_3)$-free. The 1-vertex extensions of $2P_1 + P_2$ in $\mathcal{S}$ are $3P_1 + P_2$, $P_1 + 2P_2, 2P_1 + P_3, P_1 + P_4, P_2 + P_3$ and $S_{1,1,2}$, none of which are $(K_{1,3}, 4P_1, P_1 + P_4, P_1 + 2P_2, P_2 + P_3)$-free. We conclude that $\overline{H_1} = 2P_1 + P_2$. By Theorem 76.3(i), it follows that $H_2 \in \mathcal{S}$. By Theorem 76.3(iv), it follows that H_2 is $(K_{1,3}, P_2 + P_4)$-free. By Theorem 76.4(i), it follows that H_2 is $(P_1 + P_6)$-free. The 1-vertex extensions of P_6 that are in $\mathcal{S}$ are $P_1 + P_6$, P_7, $S_{1,1,4}$ and $S_{1,2,3}$, none of which are $(K_{1,3}, P_2 + P_4, P_1 + P_6)$-free. We conclude that $H_2 = P_6$. Therefore, we do not need to consider any extensions of $(\overline{2P_1 + P_2}, P_6)$-free graphs, apart from the class of

$(\overline{2P_1 + P_2}, P_6)$-free graphs itself, and this is listed in Open Problem 2.(i).

Now consider graphs H_1, H_2 with $2P_2 \subseteq_i H_1, \overline{H_2}$ such that the class of (H_1, H_2)-free atoms has bounded clique-width, but Theorem 76 does not state that (H_1, H_2)-free atoms have unbounded clique-width. By Theorem 76.3(i) and Theorem 3(ii), respectively, H_1 and $\overline{H_2}$ must both be in $\mathcal{S}$. By Theorem 76.3(iii), it follows that H_1 and $\overline{H_2}$ are $K_{1,3}$-free, so they are both linear forests. By Theorem 76.3(vi), H_1 and $\overline{H_2}$ are $4P_1$-free, and because they are bipartite, this means they each contain at most six vertices. Since H_1 and $\overline{H_2}$ are linear forests on at most six vertices containing an induced $2P_2$, it follows that $H_1, \overline{H_2} \in \{2P_2, P_1+2P_2, P_2+P_3, P_5, 2P_1+2P_2, 3P_2, P_1+P_2+P_3, P_2+P_4, P_1+P_5, 2P_3, P_6\}$. Since H_1 and $\overline{H_2}$ are $4P_1$-free, it follows that $H_1, \overline{H_2} \in \{2P_2, P_1+2P_2, P_2+P_3, P_5, 3P_2, P_2+P_4, P_6\}$. By Theorem 76.4(v), $\overline{H_2}$ is $(P_2+P_4, 3P_2, P_5)$-free, and so $\overline{H_2} \in \{2P_2, P_1+2P_2, P_2+P_3\}$. Now if $\overline{H_2} = 2P_2$, then by Theorem 76.2(i), we may assume that H_1 is not an induced subgraph of P_6, so $H_1 \in \{P_1 + 2P_2, 3P_2, P_2 + P_4\}$ and these cases are listed in Open Problem 2.(ii). Otherwise, $\overline{H_2} \in \{P_1+2P_2, P_2+P_3\}$. In this case by Theorem 76.4(iv) H_1 is (P_1+2P_2, P_6)-free, so $H_1 \in \{2P_2, P_2 + P_3, P_5\}$. If $\overline{H_2} = P_1 + 2P_2$ then $H_1 \in \{2P_2, P_2 + P_3, P_5\}$ and these cases are listed in Open Problem 2.(iii). If $\overline{H_2} = P_2 + P_3$ then by Theorem 76.2(ii), H_1 is not an induced subgraph of $2P_2$, so $H_1 \in \{P_2 + P_3, P_5\}$ and these cases are listed in Open Problem 2.(iv). □

5.7 Conclusions

Motivated by algorithmic applications, we determined a new class of (H_1, H_2)-free graphs of unbounded clique-width whose atoms have *bounded* clique-width, namely when $(H_1, H_2) = (2P_2, \overline{P_2 + P_3})$ (in fact, our proof for $(2P_2, \overline{P_2 + P_3})$-free atoms also works for *linear* clique-width). We also identified a number of classes of (H_1, H_2)-free graphs of unbounded clique-width whose atoms still have *unbounded* clique-width. In particular, our results show that boundedness of clique-width of (H_1, H_2)-free atoms does not necessarily imply boundedness of clique-width of $(\overline{H_1}, \overline{H_2})$-free atoms. For example, (C_4, P_5)-free atoms have bounded clique-width [101], but we proved that $(\overline{C_4}, \overline{P_5})$-free atoms have unbounded clique-width (Lemma 74).

We also presented a summary theorem (Theorem 7), from which we deduced a list of **18** remaining cases of pairs (H_1, H_2) for which we do not know whether the clique-width of (H_1, H_2)-free atoms is bounded; see also Open Problem 2 and Theorem 77. In particular, we ask whether boundedness of clique-width of $(2P_2, \overline{P_2 + P_3})$-free atoms can be extended to $(P_5, \overline{P_2 + P_3})$-free atoms. Is boundedness of clique-width the underlying reason why Coloring is polynomial-time solvable on $(P_5, \overline{P_2 + P_3})$-free graphs [153]? Brandstädt and Hoàng [32] showed that $(P_5, \overline{P_2 + P_3})$-free atoms with no dominating vertices and no vertex pairs $\{x, y\}$ with $N(x) \subseteq N(y)$ are either isomorphic to some specific graph G^* or all their induced C_5s are dominating. Recently, Huang and Karthick [127] proved a more refined decomposition. However, it is not clear how to use these results to prove boundedness of clique-width of $(P_5, \overline{P_2 + P_3})$-free atoms, and additional insights are needed.

6 Vertex Deletion into Bipartite Permutation Graphs

We study the parameterized complexity of the bipartite permutation vertex deletion problem, which asks, for a given n-vertex graph, whether we can remove at most k vertices to obtain a bipartite permutation graph. A permutation graph can be defined as an intersection graph of segments whose endpoints lie on two parallel lines ℓ_1 and ℓ_2, one on each. A bipartite permutation graph is a permutation graph which is bipartite. Equivalently, the class of permutation bipartite graphs can be defined by a set of forbidden induced subgraphs [94], see Figure 6.3, which is useful in designing algorithms.

Bipartite Permutation Vertex Deletion Problem	
Instance:	A graph G and a number k
Question:	Can G be transformed into a bipartite permutation graph by performing $\leqslant k$ vertex deletions?

This problem is NP-complete by the classical result of Lewis and Yannakakis [144]. We show that it is in FPT parameterized by the number of deleted vertices. We restate our main theorem here.

Theorem 8. *There is an $\mathcal{O}(9^k \cdot |V(G)|^9)$-time algorithm for instances (G, k) of the problem of vertex deletion into bipartite permutation graphs.*

As mentioned in Chapter 1, the first (and main) step to prove Theorem 8 includes a structural analysis of the so-called *almost bipartite permutation* graphs—graphs with the same set of forbidden induced subgraphs as bipartite permutation graphs with an exception that they may contain holes on more than ten vertices in contrast to bipartite permutation graphs. Except the characterization by forbidden induced subgraphs, we use a characterization of bipartite permutation graphs by Spinrad, Brandstädt, and Stewart [166], who showed that the vertices of every connected bipartite permutation graph $G = (U, W, E)$ can be embedded into a strip in such a way that the vertices from U are on the bottom edge of the strip, the vertices from W are on the top edge of the strip, the neighbors $N(u)$ of u occur consecutively on the top edge of the strip for every $u \in U$ (adjacency property), the vertices from $N(u) - N(u')$ occur consecutively on the top edge of the strip for every $u, u' \in U$ (enclosure property), and the analogous properties are satisfied by the vertices in W (see Figure 1.3).

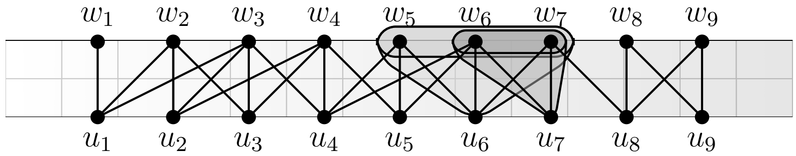


Figure 6.1 Embedding of a bipartite permutation graph (U, W, E) into a strip satysfying the adjacency and the enclosure properties.

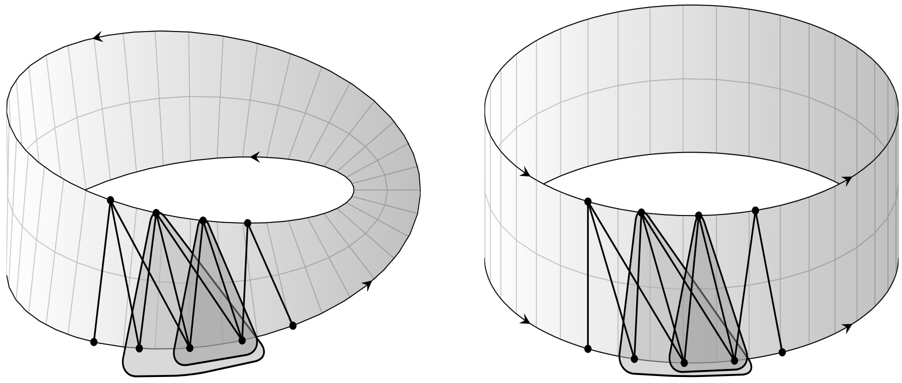


Figure 6.2 An embedding of a connected almost bipartite permutation graph in a cylinder or a Möbius strip that locally satisfies the adjacency and enclosure properties.

We characterized the structure of almost bipartite permutation graphs similarly. Informally, the class of almost bipartite permutation graphs can be embedded in such a strip as well but the ends of the strip "are glued together" to create either a cylinder, or a Möbius strip (see Figure 6.2). The formal description is given in our main structural theorem, Lemma 85. We use the developed structure to obtain an algorithm for the bipartite permutation vertex deletion problem with running time $\mathcal{O}(9^k \cdot n^9)$ Theorem 8.

6.1 Preliminaries

For a graph $G = (V, E)$, a pair $(V, <)$ is a *transitive orientation* of G if $<$ is a transitive and irreflexive relation on V that satisfies either $u < v$ or $v < u$ iff $uv \in E$ for every $u, v \in V$.

A *partially ordered set* (shortly *partial order* or *poset*) is a pair $P = (X, \leqslant_P)$ that consists of a set X and a reflexive, transitive, and antisymmetric relation $\leqslant_P$ on X. For a poset $(X, \leqslant_P)$, let the *strict partial order* $<_P$ be a binary relation defined on X such that $x <_P y$ if and only if $x \leqslant_P y$ and $x \neq y$. Equivalently, $(X, <_P)$ is a strict partial order if $<_P$ is irreflexive and transitive. Two elements $x, y \in X$ are *comparable* in P if $x \leqslant_P y$ or $y \leqslant_P x$; otherwise, x, y are *incomparable* in P. A *linear order* $L = (X, \leqslant_L)$ is a partial order in which every two vertices $x, y \in X$ are comparable. A *strict linear order* $(X, <_L)$ is a binary relation defined in a way that $x <_L y$ if and only if $x \leqslant_L y$ and $x \neq y$.

Let $P = (X, \leqslant_P)$ be a poset. A linear order $L = (X, \leqslant_L)$ is called a *linear extension* of P if $\leqslant_P \subseteq \leqslant_L$. Given a family of posets $\mathcal{P} = \{P_i = (X, \leqslant_{P_i}) : i \in I\}$, we say that P is the *intersection of* $\mathcal{P}$ if for every $x, y \in X$ we have $x \leqslant_P y$ if and only if $x \leqslant_{P_i} y$ for every $i \in I$. The *dimension* of a poset P is the minimal number of linear extensions of P that intersect to P. In particular, we say that P is *two-dimensional* if it is the intersection of two linear extensions of P.

A *comparability graph* (*incomparability graph*) of a poset $P = (X, \leqslant_P)$ has X as the set of its vertices and the set including every two vertices comparable (incomparable, respectively) in P as the set of its edges. Note the following: if $(X, \leqslant_P)$ is a poset, then $(X, <_P)$ is a transitive orientation of the comparability graph of P. A graph $G = (V, E)$ is a *comparability graph* (*co-comparability graph*) if G is a comparability (incomparability, respectively) graph of some poset defined on V. So, G is a comparability graph if and only if G admits a transitive orientation. A graph G is a *permutation graph* if and only if G and the complement of G are comparability graphs [160] (or equivalently, G and the complement of G admit transitive orientations). Baker, Fishburn, and Roberts [12] proved that G is a permutation graph if and only if G is the incomparability graph of a two-dimensional poset.

We say that two sets X and Y are *comparable* if X and Y are comparable with respect to $\subseteq$-relation (that is, $X \subseteq Y$ or $Y \subseteq X$ holds). We use the convenient notation $[m] := \{0, 1, \ldots, m\}$, for every $m \in \mathbb{N}$. For every $i, j \in \mathbb{Z}$ such that $i \leqslant j$ by $[i, j]$ we mean the set $\{i, i+1, \ldots, j\}$.

6.2 The Structure of (Almost) Bipartite Permutation Graphs

The characterization of bipartite permutation graphs presented below was proposed by Spinrad, Brandstädt, and Stewart [166].

Suppose $G = (U, W, E)$ is a connected bipartite graph. A linear order $(W, <_W)$ satisfies *adjacency property* if for each vertex $u \in U$ the set $N(u)$ consists of vertices that are consecutive in $(W, <_W)$. A linear order $(W, <_W)$ satisfies *enclosure property* if for every pair of vertices $u, u' \in U$ such that $N(u)$ is a subset of $N(u')$, vertices in $N(u') - N(u)$ occur consecutively in $(W, <_W)$. A *strong ordering* of the vertices of $U \cup W$ consists of linear orders $(U, <_U)$ and $(W, <_W)$ such that for every $(u, w'), (u', w) \in E$, where $u, u' \in U$, and $w, w' \in W$, it holds that $u <_U u'$ and $w <_W w'$ imply $(u, w) \in E$ and $(u', w') \in E$. Note that, whenever $(U, <_U)$ and $(W, <_W)$ form a strong ordering of $U \cup W$, then $(U, <_U)$ and $(W, <_W)$ satisfy the adjacency and enclosure properties.

Theorem 78 (Spinrad, Brandstädt, Stewart [166])**.** *The following three statements are equivalent for a connected bipartite graph $G = (U, W, E)$:*

(a) (U, W, E) *is a bipartite permutation graph.*

(b) There exists a strong ordering of $U \cup W$*.*

(c) There exists a linear order $(W, <_W)$ *of* W *satisfying adjacency and enclosure properties.*

An example of a bipartite permutation graph $G = (U, W, E)$ with linear order $w_1 <_W w_2 <_W \ldots <_W w_8 <_W w_9$ of the vertices of W which satisfies the adjacency and the enclosure properties is shown in Figure 6.1.

Another characterization of bipartite permutation graphs can be obtained by listing all minimal forbidden induced subgraphs for this class of graphs. Such a list can be compiled by taking all odd cycles of length $\geqslant 3$ (forbidden structures for bipartite graphs) and all bipartite graphs from the list of forbidden structures for permutation graphs obtained by Gallai [94]. The whole list is shown in Figure 6.3.

6.2.1 Almost Bipartite Permutation Graphs

The goal of this section is to characterize graphs which do not contain small forbidden subgraphs for the class of bipartite permutation graphs. Following terminology of van 't Hof and Villanger [123] we call such graphs almost bipartite permutation graphs.

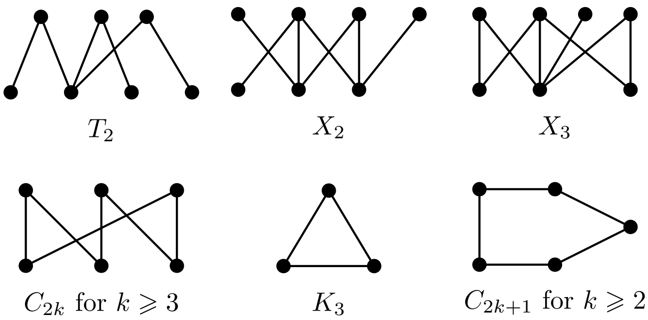


Figure 6.3 Forbidden structures for bipartite permutation graphs.

Definition 79. A graph $G = (V, E)$ is *almost bipartite permutation graph* if G does not contain T_2, X_2, X_3, K_3, C_k for $k \in [5, 9]$ as induced subgraphs.

Suppose $G = (V, E)$ is a connected almost bipartite permutation graph.

Proposition 80. *Every hole in G is a dominating set.*

Proof. Let $C = \{c_0, c_1, \dots, c_{m-1}\}$ be a hole in G. Hence, $m \geqslant 10$. Suppose, for contradiction, that there exists a vertex in the set $V \setminus (C \cup N(C))$. As G is connected, there must exist $v \in V$ at distance two from C. Let $w \in N(v) \cap N(C)$ and let c_j be a neighbor of w in C. We now look at the neighborhood of w. As G contains no triangle, wc_{j-1} and wc_{j+1} are non-edges. Moreover, as G contains no copy of T_2, vertex w is adjacent to at least one of c_{j-2} and c_{j+2}, say c_{j-2}. Thus, w is nonadjacent to c_{j-3}. Therefore, the set $\{c_{j-3}, c_{j-2}, c_{j-1}, c_j, c_{j+1}, w, v\}$ induces a copy of X_2 in G, which leads to a contradiction. □

Let C be a shortest hole in G, m be the size of C, and $c_0, c_1, \dots, c_{m-1}$ be the consecutive vertices of C, $m \geqslant 10$. In the remaining part of this chapter we use the following notation with respect to C. For any integral number i by c_i we denote the unique vertex $c_{i \bmod m}$ from the cycle C. For any two different vertices c_i, c_j in C, by *the set of all vertices between c_i and c_j from C* we mean the set $\{c_i, c_{i+1}, \dots, c_{i+k}\}$, where k is the smallest natural number such that $c_{i+k} = c_j$. Note that this notion is not symmetric, i.e., the set of all vertices between c_j and c_i from C contains c_i, c_j and all the vertices from C that are not between c_i and c_j.

Proposition 81. *For every vertex $v \in V$ either:*

(1) $N(v) \cap C = \{c_i\}$ for some $i \in [m-1]$, or

(2) $N(v) \cap C = \{c_i, c_{i+2}\}$ for some $i \in [m-1]$.

Proof. Since C is an induced cycle, (2) clearly holds for the vertices from C, so let v be a vertex in $V \setminus C$. As C is a dominating set, by Proposition 80, vertex v has at least one neighbor in C. If v has exactly one neighbor in C, then (1) holds and we are done. So assume that it has more than one neighbor. We now distinguish two cases. First, suppose that there exist two vertices $c_j, c_\ell \in N(v) \cap C$ at distance at least three in C such that v has no neighbor in the set of vertices between c_j and c_l, except c_j and c_l. Then,

$\{c_j, c_{j+1}, \ldots, c_\ell, v\}$ induces a cycle C' on at least five vertices in G. As c_j and c_ℓ are at distance at least three in C, C' is shorter than C. In particular, C' contradicts either G containing no copy of C_ℓ, for $\ell \in \{5, \ldots, 9\}$, or C being a shortest hole in G. Therefore, this case never occurs.

Hence, v has either (i) exactly two neighbors in C and those are at distance two as there is no triangle in G, so (2) holds, or (ii) C has an even number of vertices and v is adjacent to every second vertex of C. It remains to show that the latter never occurs. Indeed, if it does, then without loss of generality $c_0 \in N(v)$. But observe that since C has at least ten vertices, the set $\{c_0, c_1, c_2, c_3, c_4, c_6, v\}$ induces a copy of X_3. This concludes the proof. $\square$

Given Proposition 81, for every $i \in [m-1]$ we can set $A_i = \Big\{v \in V : N(v) \cap C = \{c_{i-1}, c_{i+1}\}\Big\}$ and $B_i = \Big\{v \in V : N(v) \cap C = \{c_i\}\Big\}$. Note that sets $A_0, B_0, \ldots, A_{m-1}, B_{m-1}$ form a partition of V. Moreover, for every $i \in [m-1]$ we have $c_i \in A_i$. Following our notation, for any integer i by A_i and B_i we denote the sets $A_{i \bmod m}$ and $B_{i \bmod m}$, respectively. Furthermore, for every $i \leqslant j$ we set:

$$A_G[i,j] = \begin{cases} A_i \cup B_{i+1} \cup A_{i+2} \cup B_{i+3} \cup \ldots \cup A_{j-1} \cup B_j & \text{if } j-i \text{ is odd,} \\ A_i \cup B_{i+1} \cup A_{i+2} \cup B_{i+3} \cup \ldots \cup B_{j-1} \cup A_j & \text{if } j-i \text{ is even,} \end{cases}$$

$$B_G[i,j] = \begin{cases} B_i \cup A_{i+1} \cup B_{i+2} \cup A_{i+3} \cup \ldots \cup B_{j-1} \cup A_j & \text{if } j-i \text{ is odd,} \\ B_i \cup A_{i+1} \cup B_{i+2} \cup A_{i+3} \cup \ldots \cup A_{j-1} \cup B_j & \text{if } j-i \text{ is even,} \end{cases}$$

$$\text{and} \quad V_G[i,j] = A_G[i,j] \cup B_G[i,j].$$

We write just $A[i,j]$, $B[i,j]$, and $V[i,j]$, respectively, instead of $A_G[i,j]$, $B_G[i,j]$, and $V_G[i,j]$, when there is no confusion.

We now characterize the neighborhoods of the vertices in sets A_i and B_i, see also Figure 6.4.

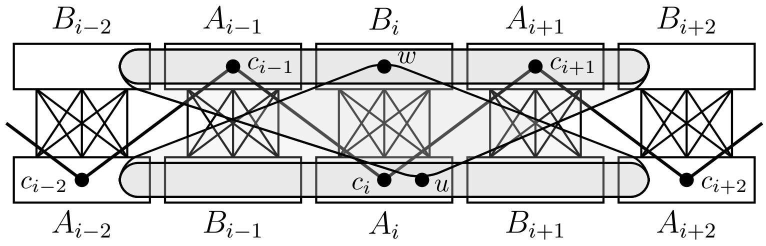


Figure 6.4 A possible neighborhood of u in A_i and w in B_i.

Proposition 82. *Let $i \in [m-1]$. Then:*

(1) A_i and B_i are independent sets.

(2) For every $u \in A_i$ and every $w \in B_i$ we have $uw \in E$.

(3) For every $u \in A_i$ we have $B_i \subseteq N(u) \subseteq B[i-2, i+2]$.

(4) For every $w \in B_i$ we have $A_i \subseteq N(w) \subseteq A[i-2, i+2]$.

Proof. Statement (1) follows trivially from the fact that G contains no triangle. To show statement (2), assume for a contrary that $uw \notin E$ for some $u \in A_i$ and some $w \in B_i$. Since $u \in A_i$, we have $N(u) \cap C = \{c_{i-1}, c_{i+1}\}$, and since $w \in B_i$ we have $N(w) \cap C = \{c_i\}$. Hence, the set $\{c_{i-2}, u, c_i, c_{i+2}, c_{i-1}, w, c_{i+1}\}$ induces an X_2 in G, which cannot be the case.

To prove statements (3) and (4), consider a graph G induced by the set $U \cup W$, where

$$U = A[i-2, i+2] \quad \text{and} \quad W = B[i-2, i+2].$$

Since any edge with two endpoints in U (or two endpoints in W) could be extended by some vertices from $\{c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}\}$ to an odd cycle of size $\leqslant 7$ in G, the graph $G[U \cup W]$ is bipartite with bipartition classes U and W.

To see that (3) holds, first note that $B_i \subseteq N(u)$ by statement (2). Therefore, suppose that u has a neighbor v in the set $V \setminus (U \cup W)$.

Consider the case when $v \in A_j$ for some $j \notin [i-2, i+2]$. Since $N(u) \cap C = \{c_{i-1}, c_{i+1}\}$ and $N(v) \cap C = \{c_{j-1}, c_{j+1}\}$, u, v and the vertices between c_{j+1} and c_{i-1} in C as well as u, v and the vertices between c_{i+1} and c_{j-1} in C induce cycles in G of size $\leqslant m-2$. Since $|C| \geqslant 10$, at least one from these cycles has size $\geqslant 6$, and as such it can not occur in G.

So suppose $v \in B_j$ for some $j \notin [i-2, i+2]$. Since $N(v) \cap C = \{c_j\}$, u, v and the vertices between c_{i+1} and c_j in C as well as u, v and the vertices between c_j and c_{i-1} in C induce holes in G of size $\leqslant m-2$, and as such they can not occur in G. So, $N(u) \subseteq W$, which completes the proof of statement (3).

Statement (4) is proved by similar arguments. □

Proposition 82 asserts that all the neighbors of the vertices from A_i and from B_i are contained in the set $B[i-2, i+2]$ and $A[i-2, i+2]$, respectively. The next proposition describes the relations that hold between the neighborhoods of the vertices from $B[i-2, i+2]$ restricted to the set A_i and between the neighborhoods of the vertices from $A[i-2, i+2]$ restricted to the set B_i.

Proposition 83. *Let $i \in [m-1]$. For $(i \pm 2, i \pm 1) \in \{(i-2, i-1), (i+2, i+1)\}$, the following hold:*

(1) For every $w, w' \in B_{i\pm2} \cup A_{i\pm1}$ the sets $N(w) \cap A_i$ and $N(w') \cap A_i$ are comparable.
Moreover, if $w \in B_{i\pm2}$ and $w' \in A_{i\pm1}$, then $N(w) \cap A_i \subseteq N(w') \cap A_i$.

(2) For every $u, u' \in A_{i\pm2} \cup B_{i\pm1}$ the sets $N(u) \cap B_i$ and $N(u') \cap B_i$ are comparable.
Moreover, if $u \in A_{i\pm2}$ and $u' \in B_{i\pm1}$, then $N(u) \cap B_i \subseteq N(u') \cap B_i$.

Proof. To prove (1), we consider the case $(i \pm 2, i \pm 1) = (i-2, i-1)$, as the other one follows by symmetry. Suppose that $w, w' \in B_{i-2} \cup A_{i-1}$ are such that neither $N(w) \cap A_i \subseteq N(w') \cap A_i$ nor $N(w') \cap A_i \subseteq N(w) \cap A_i$ holds. It means that there are $u, u' \in A_i$ such that $wu \in E$, $w'u' \in E$, $wu' \notin E$, and $w'u \notin E$. Since $w, w' \in B_{i-2} \cup A_{i-1}$, we have $c_{i-2}w, c_{i-2}w' \in E$ and $c_{i-4}w, c_{i-3}w, c_{i-4}w', c_{i-3}w' \notin E$. Furthermore, $ww' \notin E$ and $uu' \notin E$ as G contains no triangle. Consequently, the set $\{c_{i-3}, w, w', c_{i-4}, c_{i-2}, u, u'\}$ induces a copy of T_2 in G, which cannot be the case. Moreover, if $w \in B_{i-2}$, $w' \in A_{i-1}$, then since $c_i \in (N(w') \cap A_i) \setminus (N(w) \cap A_i)$, the latter statement holds.

To show (2), we again only consider the case $(i \pm 2, i \pm 1) = (i-2, i-1)$. Suppose that $u, u' \in A_{i-2} \cup B_{i-1}$ are such that neither $N(u') \cap B_i \subseteq N(u) \cap B_i$ nor $N(u) \cap B_i \subseteq N(u') \cap B_i$ holds. It means that there are $w, w' \in B_i$ such that $uw, u'w' \in E$ and $u'w, uw' \notin E$. Since $u, u' \in A_{i-2} \cup B_{i-1}$, we have $uc_{i-1}, u'c_{i-1} \in E$ and $uc_{i+1}, u'c_{i+1} \notin E$. Furthermore, $uu' \notin E$ and $ww' \notin E$ as G contains no triangle. Hence, the set $\{c_{i-1}, w, w', c_{i+1}, u, c_i, u'\}$ induces a copy of X_3 in G, which cannot be the case.

To see the second part of the statement, assume that $N(u) \cap B_i \nsubseteq N(u') \cap B_i$ for some $u \in A_{i-2}$, $u' \in B_{i-1}$. That is, there is $w \in B_i$ such that $uw \in E$ and $u'w \notin E$. In particular, it means that $u \neq c_{i-2}$. Note that $uc_{i-1}, u'c_{i-1} \in E$. Consequently, the set $\{c_{i-3}, c_{i-2}, c_{i-1}, c_i, u, u', w\}$ induces a copy of X_3 in G, which is a contradiction. □

Proposition 83 allows us to order vertices of A_i based on two properties. We now define relation $<_{A_i}$ which combines them and we show that $<_{A_i}$ is a partial order (see Figure 6.5 for an illustration). We define for every $u, u' \in A_i$:

$$u <_{A_i} u' \text{ iff } \begin{array}{l} \text{there is } w \in B_{i-2} \cup A_{i-1} \text{ such that } u \in N(w) \text{ and } u' \notin N(w), \text{ or} \\ \text{there is } w \in A_{i+1} \cup B_{i+2} \text{ such that } u' \in N(w) \text{ and } u \notin N(w), \end{array}$$

Similarly, we define a relation $<_{B_i}$ for every $w, w' \in B_i$:

$$w <_{B_i} w' \text{ iff } \begin{array}{l} \text{there is } u \in A_{i-2} \cup B_{i-1} \text{ such that } w \in N(u) \text{ and } w' \notin N(u), \text{ or} \\ \text{there is } u \in B_{i+1} \cup A_{i+2} \text{ such that } w' \in N(u) \text{ and } w \notin N(u). \end{array}$$

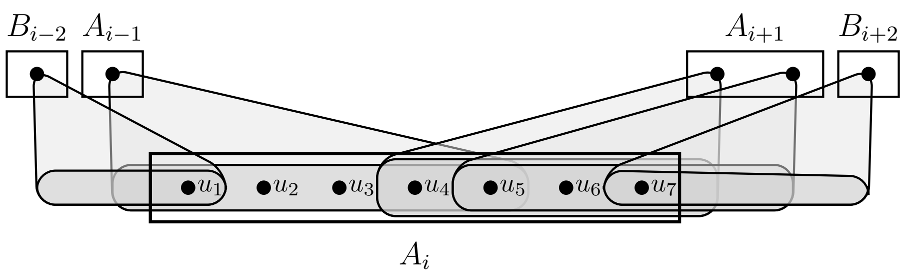


Figure 6.5 The neighborhoods of the vertices from $B_{i-2} \cup A_{i-1} \cup A_{i+1} \cup B_{i+2}$ restricted to A_i. We have $u_1 <_{A_i} \{u_2, u_3\} <_{A_i} u_4 <_{A_i} u_5 <_{A_i} u_6 <_{A_i} u_7$.

Proposition 84. *The following statements hold for every $i \in [m-1]$:*

(1) $(A_i, <_{A_i})$ is a strict partial order. Moreover, $u, u' \in A_i$ are incomparable in $(A_i, <_{A_i})$ if and only if $N(u) = N(u')$.

(2) $(B_i, <_{B_i})$ is a strict partial order. Moreover, $w, w' \in B_i$ are incomparable in $(B_i, <_{B_i})$ if and only if $N(w) = N(w')$.

Proof. Let $i \in [m-1]$ be fixed. To prove that $(A_i, <_{A_i})$ is a strict partial order, we need to show that $<_{A_i}$ is irreflexive and transitive. The irreflexivity follows from the definition, in aim to show transitivity, we first prove that $<_{A_i}$ is antisymmetric. Suppose for a contrary that there are $u, u' \in A_i$ such that $u <_{A_i} u'$, and $u' <_{A_i} u$. Suppose $u <_{A_i} u'$ is witnessed by a vertex $w \in B_{i-2} \cup A_{i-1}$ such that $u \in N(w)$ and $u' \notin N(w)$; the other case $w \in A_{i+1} \cup B_{i+2}$ is analogous. By Proposition 83.(1), there is no $w' \in B_{i-2} \cup A_{i-1}$ such that $u' \in N(w')$ and $u \notin N(w')$. Hence, since $u' <_{A_i} u$, there must be a vertex $w' \in A_{i+1} \cup B_{i+2}$ such that $u \in N(w')$ and $u' \notin N(w')$. We have $uc_{i+1}, u'c_{i+1} \in E$ and $uc_{i+2}, u'c_{i+2}, uc_{i+3}, u'c_{i+3} \notin E$ as $\{u, u'\} \subseteq A_i$. We have also $wc_{i+1}, wc_{i+2}, wc_{i+3}, w'c_{i+1}, w'c_{i+3} \notin E$ and $w'c_{i+2} \in E$ as $w \in B_{i-2} \cup A_{i-1}$ and $w' \in A_{i+1} \cup B_{i+2}$. Moreover, $uu', ww' \notin E$, by Proposition 82. Consequently, the set $\{w, w', c_{i+1}, c_{i+3}, u, u', c_{i+2}\}$ induces a copy of X_2 in G, which cannot be the case.

To show transitivity, suppose for a contrary that there are vertices $u, u', u'' \in A_i$ such that $u <_{A_i} u'$ and $u' <_{A_i} u''$, but $u <_{A_i} u''$ does not hold. Suppose $u <_{A_i} u'$ is witnessed by a vertex $w \in B_{i-2} \cup A_{i-1}$ such that $u \in N(w)$ and $u' \notin N(w)$; the other case $w \in B_{i+1} \cup A_{i+2}$ is symmetric. We have $u'' \in N(w)$ as otherwise $u <_{A_i} u''$, by definition. Suppose $u' <_{A_i} u''$ is witnessed by a vertex $w' \in B_{i-2} \cup A_{i-1} \cup A_{i+1} \cup B_{i+2}$. Note that if $w' \in B_{i-2} \cup A_{i-1}$, then $u' \notin N(w')$ and $u' \in N(w')$, which enforces also $u \in N(w')$ as $u <_{A_i} u'$ and we already proved that $<_{A_i}$ is antisymmetric. Thus, $u \in N(w')$ and $u'' \notin N(w')$, which shows $u <_{A_i} u''$. Hence, we must have $w' \in A_{i+1} \cup B_{i+2}$, and so $u' \notin N(w')$ and $u'' \in N(w')$. Moreover, $u \in N(w')$ as otherwise $u <_{A_i} u''$. As $\{u', u''\} \subseteq A_i$, $w \in A_{i-2} \cup B_{i+1}$, and $w' \in A_{i+1} \cup B_{i+2}$, we have $uu', ww' \notin E$, by Proposition 82. Consequently, $\{w, w', c_{i+1}, c_{i+3}, u', u'', c_{i+2}\}$ induces a copy of X_2 in G, which is not possible. We conclude that $(A_i, <_{A_i})$ is a strict partial order.

By definition, if $N(u) = N(u')$, then u and u' are incomparable in $(A_i, <_{A_i})$. Hence, for the second statement of (1), it is enough to show that $N(u) \neq N(u')$ implies that u and u' are comparable in $(A_i, <_{A_i})$. Let w be a vertex such that $wu \in E$ and $wu' \notin E$. By Proposition 82.(2) and (3), $w \in B_{i-2} \cup A_{i-1} \cup A_{i+1} \cup B_{i+2}$. However, if $w \in B_{i-2} \cup A_{i-1}$ then $u <_{A_i} u'$ and if $w \in A_{i+1} \cup B_{i+2}$ then $u' <_{A_i} u$, by definition. Hence, u and u' are comparable in $<_{A_i}$.

The proof of (2) is similar. For antisymmetry, suppose that we have $w, w' \in B_i$ such that $w <_{B_i} w'$ and $w' <_{B_i} w$. Let $w <_{B_i} w'$ and $w' <_{B_i} w$ be witnessed by u and u' from $A_{i-2} \cup B_{i-1} \cup B_{i+1} \cup A_{i+2}$, respectively. Analogously to (1), by Proposition 83.(2), we can assume that $u \in A_{i-2} \cup B_{i-1}$ and $u' \in B_{i+1} \cup A_{i+2}$ and $uw, u'w \in E$, $uw', u'w' \notin E$. Observe that the set $\{c_{i-1}, w, w', c_{i+1}, u, c_i, u'\}$ induces a copy of X_3 in G, a contradiction.

For transitivity of $<_{B_i}$, suppose that for some w, w', and $w'' \in B_i$ we have $w <_{B_i} w'$ and $w' <_{B_i} w''$, but $w <_{B_i} w''$ does not hold. By symmetry of the proof of (1), we reach the case $u \in A_{i-2} \cup B_{i-1}$ and $u' \in B_{i+1} \cup A_{i+2}$, $uw, uw'', u'w, u'w'' \in E$ and $uw', u'w' \notin E$. Then, one can easily check that the set $\{w', w'', c_{i+1}, u, c_i, u', c_{i+2}\}$ induces a copy of X_2 in G, a contradiction.

Now, assume that $N(w) \neq N(w')$. Without loss of generality assume that there exists $u \in A_{i-2} \cup B_{i-1} \cup B_{i+1} \cup A_{i+2}$ such that $u \in N(w) \setminus N(w')$. Analogously as before, observe that if $u \in A_{i-2} \cup B_{i-1}$ then $w <_{B_i} w'$ and if $u \in B_{i+1} \cup A_{i+2}$ then $w' <_{B_i} w$. Therefore, w and w' are comparable, which finishes the proof. □

Finally, for every $i \in [m-1]$ we order arbitrarily the elements inside every antichain of $(A_i, <_{A_i})$ and of $(B_i, <_{B_i})$, obtaining strict linear orders $(A_i, <_{A_i})$ and $(B_i, <_{B_i})$. We introduce a binary relation $\prec$ defined on the set V, such that $v \prec v'$ for $v, v' \in V$ if one of the following conditions holds for some $i \in [m-1]$:

- $v, v' \in A_i$, $v <_{A_i} v'$, and v, v' are consecutive in $(A_i, <_{A_i})$,
- $v, v' \in B_i$, $v <_{B_i} v'$, and v, v' are consecutive in $(B_i, <_{B_i})$,
- v is the maximum of $(A_i, <_{A_i})$ and v' is the minimum of $(B_{i+1}, <_{B_{i+1}})$,
- v is the maximum of $(B_i, <_{B_i})$ and v' is the minimum of $(A_{i+1}, <_{A_{i+1}})$.

Informally, to get an embedding of G into a cylinder (the shortest hole is even) or into a Möbius strip (the shortest hole is odd) which locally satisfies the adjacency and the enclosure properties, we place the vertices v, v' satisfying $v \prec v'$ next to each other, v before v' assuming that the border of the cylinder or the Möbius strip are oriented as shown in Figure 6.2. In what follows we extend $\prec$ relation as follows:

- For every $V' \subsetneq V$ by $<_{V'}$ we denote the transitive closure of $\prec$ restricted to V',
- For $v, v' \in V$ we set $v <_{cl} v'$ if $v, v' \in A[i-2, i+2]$ and $v <_{A[i-2,i+2]} v'$ for some $i \in [m-1]$ or $v, v' \in B[i-2, i+2]$ and $v <_{B[i-2,i+2]} v'$ for some $i \in [m-1]$.

Finally, we obtained the structural characterization of an almost bipartite permutation graph:

Lemma 85. *Let i, j be such that $i \leqslant j$, $|j-i| = m-3$. Let $U = A[i,j]$ and $W = B[i,j]$. Then $G[U \cup W]$ is a bipartite permutation graph with bipartition classes U and W.*

Moreover, $(U, <_U)$ and $(W, <_W)$ are strict linear orders that satisfy the adjacency and enclosure properties in $G[U \cup W]$.

Proof. Proposition 82 asserts there is no edge between a vertex in $V[j-1, j]$ and a vertex in $V[i, i+1]$. In particular, $G[U \cup W]$ is a bipartite graph and $(U, <_U)$ and $(W, <_W)$ are strict linear orders. Given Theorem 78.(c), to prove the lemma we need to show that $(W, <_W)$ satisfies the adjacency and enclosure properties in $G[U \cup W]$.

To prove the adjacency property, consider $u \in A_k \subseteq U$ for some suitable k. Recall that by Proposition 82.(3), $B_k \subseteq N(u) \subseteq B[k-2, k+2]$. To show that $N(u)$ consists of consecutive vertices in W it suffices to note that:

- if $w \in A_{k+1}$, $w' \in B_{k+2}$ and $uw' \in E$ then $uw \in E$, by Proposition 83,
- if $w, w' \in A_{k+1}$ (resp. $w, w' \in B_{k+2}$) are such that $w <_{A_{k+1}} w'$ (resp. $w <_{B_{k+2}} w'$) and $uw' \in E$, then $uw \in E$, by Proposition 84,

and that analogous statements hold by symmetry for the part of $N(u)$ contained in $A_{k-1} \cup B_{k-2}$. If $u \in B_k \subseteq U$, the case analysis is similar (one needs to swap letters A and B in the reasoning above). Therefore, the adjacency property is proved.

To show that $(W, <_W)$ satisfies the enclosure property assume for a contradiction that there are $w, w', w'' \in W$ and $u, u' \in U$ such that $N(u') \subseteq N(u)$, $w <_W w' <_W w''$ and $uw, uw', uw'' \in E$, $u'w' \in E$, and $u'w, u'w'' \notin E$.

Claim 86.86. *There is $k \in [i, j]$ such that either $u, u' \in A_k$, or $u, u' \in B_k$.*

of Claim. If $u \in B_k$, then since $N(u') \cap C \subseteq N(u) \cap C = \{c_k\}$, we have $u' \in B_k$, so the claim holds. Therefore, assume that $u \in A_k$, and suppose that $u' \notin A_k$. Then $N(u') \cap C \subseteq N(u) \cap C = \{c_{k-1}, c_{k+1}\}$. Assuming $u <_U u'$ (the other case is symmetric), we have that $u' \in B_{k+1}$. Due to Proposition 82 and $w' <_W w''$ we have $w', w'' \in A[k-1, k+2]$. Moreover, as we already proved that $N(u')$ is consecutive in $(W, <_W)$ (adjacency property), and $c_{k+1} \in N(u')$, we have $c_{k+1} <_W w''$. Therefore $w'' \in A_{k+1} \cup B_{k+2}$. Note that:

- if $w'' \in B_{k+2}$, then, since $uw'' \in E$, we have that $w'' \in N(u) \cap B_{k+2}$. However, by Proposition 83.(2), we have $N(u) \cap B_{k+2} \subseteq N(u') \cap B_{k+2}$, so it implies that $u'w'' \in E$, a contradiction,

- if $w'' \in A_{k+1}$, then by Proposition 82.(2) we would have $u'w'' \in E$, a contradiction.

This concludes the proof of claim. □

Suppose $u, u' \in A_k$. Since $u'w, u'w'' \notin E$ and $u'c_{k-1}, u'c_{k+1} \in E$, we have by adjacency property of $(W, <_W)$ that $w <_W c_{k-1} <_W w''$. Therefore, we must have $w \in B_{k-2} \cup A_{k-1}$ and $w'' \in A_{k+1} \cup B_{k+2}$. Observe that w'' witnesses that $u' <_{A_k} u$ by definition, however, w witnesses the opposite, that is $u <_{A_k} u'$. We have a contradiction by Proposition 84.

Suppose $u, u' \in B_k$. An analysis, which is analogous to the one in the previous case (again, it is enough to swap letters A and B in that reasoning above), gives us that we must have $w \in A_{k-2} \cup B_{k-1}$ and $w'' \in B_{k+1} \cup A_{k+2}$. Again, we obtain a contradiction by the definition of $<_{B_k}$ and Proposition 84. □

Lemma 85 provides an interesting view on classification of almost bipartite permutation graphs. Specifically, if m is even, then the graph may be drawn on a cylinder, whose boundary consists of two closed curves, one of which traverses vertices of $A[0, m-1]$, and the second one—the vertices of $B[0, m-1]$. If in turn m is odd, then the graph can be represented on a Möbius strip, whose boundary traverses consecutive vertices of $A[0, m-1]$ and then $B[0, m-1]$ (recall Figure 6.2).

The following definitions are taken from [123]. A *hole cut* of G is a vertex set $X \subseteq V$ such that $G - X$ is a bipartite permutation graph. Lemma 85 asserts that for every $i \in [m-1]$ the set $V[i, i+1]$ is a hole cut in G. A hole cut X of G is *minimum* if G does not have a hole cut whose size is strictly smaller than the size of X. A hole cut X of G is *minimal* if any proper subset of X is not a hole cut in G.

The next proposition describes the structure of every hole in G.

Proposition 87. *Suppose C' is a hole of size k in G for some $k \geqslant m$. Then, the consecutive vertices of C' can be labeled by $c'_0, c'_1, \ldots, c'_{k-1}$ so as the following conditions hold (the indices are taken modulo k):*

- $c'_i c'_{i+1} \in E$ *for every* $i \in [k-1]$,
- $c'_i <_{cl} c'_{i+2}$ *for every* $i \in [k-1]$,
- $\{c'' \in C' : c'_i <_{cl} c'' <_{cl} c'_{i+2}\} = \emptyset$ *for every* $i \in [k-1]$.

Proof. Let $c'_0 = v_i$, c'_1, c'_2, $\ldots$, c'_{n-1} be consecutive vertices of C' denoted in such a way that $c'_0 <_{cl} c'_2$. We can assume it, since $c'_0, c'_2 \in N(c'_1)$, thus, by Proposition 82.(3) and (4), both c'_0, c'_2 belong to $A[\ell-2, \ell+2]$ or both belong to $B[\ell-2, \ell+2]$ for some $\ell \in [m-1]$.

Now, we show that if there exists $j \in [m-1]$ such that $c'_j <_{cl} c'_{j+2}$, then $c'_{j+1} <_{cl} c'_{j+3}$. Suppose, for contradiction that $c'_j <_{cl} c'_{j+2}$ and $c'_{j+1} \not<_{cl} c'_{j+3}$. Let $i \in [m-1]$ be such that $c_{j+2} \in (A_i \cup B_i)$. Similarly, as $c'_{j+1}, c'_{j+3} \in N(c'_{j+2})$, either $c'_{j+1}, c'_{j+3} \in B[i-2, i+2]$ if $c'_{j+2} \in A_i$ or $c'_{j+1}, c'_{j+3} \in A[i-2, i+2]$ if $c'_{j+2} \in B_i$. In both cases Lemma 85 implies that $<_{cl}$ restricted to $V[i-4, i+2]$ is a strong ordering of $G[V[i-4, i+2]]$. Moreover, c'_{j+1}, c'_{j+3} are comparable in $<_{cl}$, by Proposition 82.(3) and (4), and the definition of $<_{cl}$, thus, $c'_{j+3} <_{cl} c'_{j+1}$. From Theorem 78.(b) we get that $c'_j c'_{j+3} \in E$, so C' has a chord—contradiction. Therefore $c'_j <_{cl} c'_{j+2}$ implies $c'_{j+1} <_{cl} c'_{j+3}$ for every integer j. Applying above observation repeatedly for $j = 0, 1, 2, \ldots$, we get that $c'_0 <_{cl} c'_2 <_{cl} c'_4 <_{cl} \ldots$ and $c'_1 <_{cl} c'_3 <_{cl} c'_5 <_{cl} \ldots$

For the last property, suppose for the sake of contradiction that there exists $j \notin \{i, i+2\}$ such that $c'_i <_{cl} c'_j <_{cl} c'_{i+2}$. Then, by Lemma 85, $c'_j c'_{i+1} \in E$ due to the adjacency property. But then the edge $c'_j c'_{i+1}$ is a chord in C'. This completes the proof. □

The structure of holes described above asserts that for every $i \in [m-1]$ the sets $A[i-2, i+2]$ and $B[i-2, i+2]$ are hole cuts. We use this observation to prove the following statement about minimal hole cuts in G.

Proposition 88. *Every minimal hole cut X in G is fully contained in the set $V[i-2, i+2]$ for some $i \in [m-1]$.*

Proof. First, note that we can choose elements z_1, x_1, x_2, z_2 in V and an index $i \in [m-1]$ such that the following conditions hold:

- we have $z_1 \prec x_1 \leqslant_{cl} x_2 \prec z_2$, the set $X' = \{x : x_1 \leqslant_{cl} x \leqslant_{cl} x_2\}$ is non-empty and is contained in X, and the elements z_1, z_2 are not in X.

Note that either $\{z_1, z_2\} \cup X' \subseteq B[i-2, i+2]$ or $\{z_1, z_2\} \cup X' \subseteq A[i-2, i+2]$ for some $i \in [m-1]$. Otherwise, we have $B[j, j+3] \subseteq X'$ or $A[j, j+3] \subseteq X'$ for some $j \in [m-2]$. However, by combining Proposition 87 and Proposition 82.(3) and (4), the sets $A[j, j+3]$ and $B[j, j+3]$ are hole cuts for every $j \in [m-1]$. In particular, we have either $B[j, j+3] = X' = X$ or $A[j, j+3] = X' = X$ as X is a minimal hole cut. But then, we have $X \subseteq V[j, j+3]$, which completes the proof of our claim. So, for the rest of the proof we assume $\{z_1, z_2\} \cup X' \subseteq B[i-2, i+2]$; the other case is proved similarly. Moreover, we may assume that i is picked such that:

- $z_1 \in B[i-2, i]$, $z_2 \in B[i, i+2]$, and $\{z_1, z_2\} \cup X' \subset B[i-2, i+2]$.

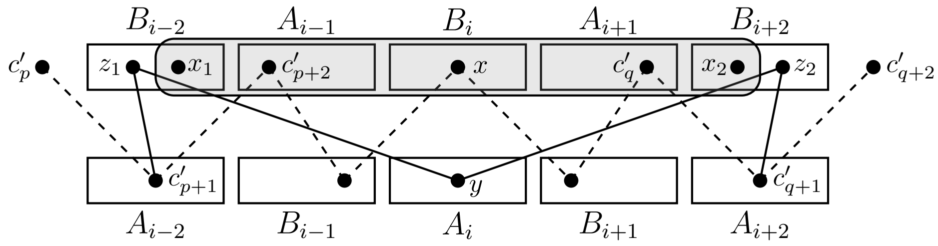


Figure 6.6 Illustration of the proof: the cycle C' is marked with a dashed line. The set X' is shaded.

See Figure 6.6 for an illustration.

Suppose Y' is the set consisting of all the neighbors of z_1 and z_2; that is, $Y' = N(z_1) \cap N(z_2)$. Clearly, we have $Y' \subset A[i-2, i+2]$. To complete the proof of the proposition we show that:

- every element of Y' is a member of X,
- $X' \cup Y'$ is a hole cut in G.

Then we have $X' \cup Y' = X$ by minimality of X and consequently $X \subset V[i-2, i+2]$. So, it remains to prove the claims about the set Y'.

Suppose we have $y \in Y'$ such that $y \notin X$. Since X is a minimal hole cut, $X \setminus \{x\}$ is not a hole cut, where x is some fixed element from X'. That is, there is a hole C' in $G - (X \setminus \{x\})$. Note that C' must contain x. Suppose $c'_0, \dots, c'_{\ell-1}$ for some $\ell \geqslant 9$ are consecutive vertices in C' chosen such that $c'_j <_{cl} c_{j+2}$ for every $j \in [\ell-1]$ (indices are taken modulo ℓ). Now we pick $p, q \in [\ell-1]$ such that $c'_p <_{cl} z_1 \leqslant_{cl} c'_{p+2}$ and $c'_q \leqslant_{cl} z_2 <_{cl} c'_{q+2}$. Since $x \in C'$, we have $c'_{p+2} \leqslant_{cl} x$ and $x <_{cl} c'_q$. Note that c'_{p+1} is adjacent to z_1 and c'_{q+1} is adjacent to z_2. Next we replace in C' all the vertices between c'_{p+2} and c'_q (this set includes x) with the vertices z_1, y, z_2 and we obtain a cycle C'' containing no elements from X. Clearly, we can easily find a hole among the elements from C'' that avoids all the elements from X. This yields a contradiction as X is a hole cut.

To prove the second claim, suppose there is a hole C' in $G - (X' \cup Y')$. By Proposition 87 there are $c'_1, c'_2, c'_3 \in C'$ such that $c'_1 <_{cl} X' <_{cl} c'_3$ and $c'_1, c'_3 \in N(c_2)$. However, since $c'_1 \leqslant_{cl} z_1 <_{cl} z_2 \leqslant_{cl} c'_3$ and $c'_1, c'_3 \in N(c'_2)$, we have $z_1 \in N(c'_2)$ and $z_2 \in N(c'_2)$. So, we have $c'_2 \in Y'$, which is a contradiction. $\square$

6.3 Proof of Theorem 8

The aim of this section is to provide a complete proof of Theorem 8 using structural results from the previous section. Let us start by showing that the Bipartite Permutation Vertex Deletion problem can be decided in polynomial time on almost bipartite permutation graphs.

Lemma 89. *Let (G, k) be an instance of* Bipartite Permutation Vertex Deletion *where G is an n-vertex almost bipartite permutation graph. Then* Bipartite Permutation Vertex Deletion *can be decided in time $\mathcal{O}(n^6)$.*

Proof. If G is a bipartite permutation graph, (G,k) is a **yes**-instance, thus, we are done in this case. If G is not connected, we can consider each connected component independently and, at the end, we compare k with the total number of deleted vertices over all components. Let G' be a connected r-vertex component of G such that G' is not a bipartite permutation graph (otherwise, clearly, no vertex needs to be deleted). Let $C = \{c_0, \ldots, c_{m-1}\}$ be a shortest hole in G' (it exists as G' is not a bipartite permutation graph). It can be found in time $\mathcal{O}(r^6)$ as follows. We iterate over all possible four-element subsets $S = \{v_1, v_2, v_3, v_4\}$ of $V(G')$. For these S for which $G'[S]$ is an induced P_4, with consecutive vertices v_1, v_2, v_3, v_4, we construct a graph $\widetilde{G'}$ by removing the vertices from $(N(v_2) \cup N(v_3)) \setminus \{v_1, v_4\}$ (note that v_2 and v_3 also get removed). Then we find a shortest v_1-v_4-path in $\widetilde{G'}$ in time $\mathcal{O}(r^2)$.

By Proposition 88, every minimal hole cut X in G' is contained in the set $V' = V_{G'}[i-2, i+2]$ for some $i \in [m-1]$. Therefore, we may check all the possibilities where a minimal cut is contained. For every i, we run an algorithm for finding a maximum flow in the following digraph H_i.

Digraph H_i has the vertex set $V' \times \{\mathsf{in}, \mathsf{out}\} \cup \{s, t\}$ and arc set consisting of:

- all arcs of the form $(u, \mathsf{out})(v, \mathsf{in})$, where uv is an edge of $G'[V']$,
- $s(v, \mathsf{in})$ if there exists $u \in V_{G'}[i-4, i-3]$ such that uv is an edge of G',
- $(u, \mathsf{out})t$ if there exists $v \in V_{G'}[i+3, i+4]$ such that uv is an edge of G',
- $(u, \mathsf{in})(u, \mathsf{out})$ for all $u \in V'$.

Set capacities of arcs of the form $(u, \mathsf{in})(u, \mathsf{out})$ to 1 and capacities of all the remaining arcs to ∞ (practically $|V_{G'}|$). It is readily seen that minimum (s,t)-cut in the defined network H_i corresponds to minimum hole cut in $G'[V']$ (arc of unit capacity $(u, \mathsf{in})(u, \mathsf{out})$ naturally corresponds to the vertex u of G').

Therefore it remains to apply classical max-flow algorithm to each H_i for $i \in [m-1]$ and remember the smallest size $k_{G'}$ of minimal (s,t)-cuts. This can be performed in time $\mathcal{O}(m \cdot (|V'| + 2) \cdot (|E_{G'[V']}| + 2|V'|)^2) = \mathcal{O}(r^6)$ [83]. Finally, (G,k) is a **yes**-instance if and only if the sum of remembered sizes $k_{G'}$ over the all considered connected components G' is at most k. Clearly, the total running time is $\mathcal{O}(n^6)$. □

We now propose the algorithm. Given an n-vertex graph $G = (V, E)$ and number k, we want to answer the Bipartite Permutation Vertex Deletion problem. We say that (G,k) is the *initial* instance. We split our algorithm into two parts. The first part consists of a branching algorithm for deletion to almost bipartite permutation graphs. The output of the first part is a set of instances (G', k') where G' is an almost bipartite permutation graph and $0 \leq k' \leq k$ (or **no**-answer is no such instance exists) such that the initial instance (G,k) is a **yes**-instance if and only if at least one of these instances is a **yes**-instance. We show that the overall time of the first phase is $\mathcal{O}(n^9 \cdot 9^k)$. In the second part, the algorithm runs an $\mathcal{O}(n^6)$-time algorithm for Bipartite Permutation Vertex Deletion for each instance (G', k') output by the first phase. In the second part, the algorithm runs an $\mathcal{O}(n^6)$ algorithm for Bipartite Permutation Vertex Deletion for each instance (G', k') output by the first phase.

Let us start with the first part. We say that $X \subseteq V$ is a *forbidden set* if $G[X]$ is isomorphic to one of the graphs: $K_3, T_2, X_2, X_3, C_5, C_6, C_7, C_8, C_9$. We define the following rule.

Rule : Given an instance (G, k), $k \geq 1$, and a minimal forbidden set X, branch into $|X|$ instances, $(G - v, k - 1)$ for each $v \in X$.

Starting with the initial instance, the algorithm applies the rule exhaustively. In other words, the algorithm is a branching tree with leaves corresponding to instances (G', k') where $k' = 0$ or G' is an almost bipartite permutation graph. Clearly, as at least one vertex from each forbidden set must be removed from G, the initial instance is a **yes**-instance if and only if at least one of the leaves is a **yes**-instance.

The algorithm continues to the second part only with such leaves (G', k') that G' is an almost bipartite permutation graph (as otherwise, the leaf is **no**-instance). It runs the algorithm described in Lemma 89 to find if G' can be transformed into a bipartite permutation graph by using at most k' vertex deletions. It either finds a **yes**-instance or concludes after checking all the instances that there is no solution; that is, the initial instance is a **no**-instance.

We note that such a branching into a bounded number of smaller instances is a standard technique, see e.g., [123] for more details.

We now analyze the running time of the whole algorithm. In the first part, observe that the branching tree has depth at most k and has at most 9^k leaves, as k decreases by one whenever the algorithm branches and each of the listed forbidden subgraphs has at most nine vertices. Therefore the total number of nodes in the branching tree is $\mathcal{O}(9^k)$. Moreover, in each node (G'', k''), the algorithm works in time $\mathcal{O}(n^9)$ as it checks if G'' contains a forbidden set. In the second part, the algorithm does a work $\mathcal{O}(n^6)$ in each leaf, by Lemma 89. We conclude that the total running time of our algorithm for Bipartite Permutation Vertex Deletion is $\mathcal{O}(9^k \cdot n^9)$.

6.4 Conclusion

We investigate for the first time the modification problems in graph classes related to partial orders. Our main result says that the bipartite permutation vertex deletion problem is fixed parameter tractable. We leave open the following two questions that inspired our research.

Open Problem 3. *What is the parameterized status of the vertex deletion problems to the class of permutation graphs and to the class of co-comparability graphs?*

We recall that, due to the result of Lewis and Yannakakis [144], both of these problems are NP-complete. One of the most important result of our work is the description of the structure of almost bipartite permutation graphs, which are defined as graphs which do not induce small graphs from the list of forbidden structures for bipartite permutation graphs. In a similar fashion we can define the class of *almost permutation* and *almost co-comparability graphs*. The next two questions seem very natural in order to solve Problem 3.

Open Problem 4. *What is the structure of almost permutation and almost co-comparability graphs?*

We are aware that the two problems mentioned above can be quite difficult. Therefore, it is worth considering intermediate problems that may be easier to attack. One of the proposed simplifications relies on the transition from the world of graphs to the world of posets. The following *vertex deletion into two-dimensional posets* problem seems very natural in the context of our research: we are given in the input a poset P and a number k and we ask whether we can delete at most k points from P so that the remaining points induce a two-dimensional poset in P.

Open Problem 5. *What is the parameterized status of the vertex deletion into two-dimensional poset problem?*

Since permutation graphs are co-comparability graphs of two-dimensional posets and since permutation graphs are both comparability and co-comparability graphs, the vertex deletion into two-dimensional poset problem is equivalent to the vertex deletion into co-comparability graph (or into permutation graph) problem if we assume that only comparability graphs can be given in the input. The class of two-dimensional posets is very well understood; in particular, the list of minimal forbidden structures for this class of posets, which is still infinite, is known (obtained independently by Trotter and Moore [169] and by Kelly [132]). Of course, it is natural to ask the following question:

Open Problem 6. *What is the structure of almost two-dimensional posets?*

Since the comparability graphs of posets do not contain odd holes of size $\geqslant 5$, we know the structure of almost two-dimensional posets that are bipartite. Indeed, these are the posets whose comparability graphs are almost bipartite permutation graphs embeddable into cylinder stripes.

7 Tuza's Conjecture for Threshold Graphs

We recall that for a graph G, a *triangle packing* is a family of pairwise edge-disjoint triangles, and a *triangle hitting* is a subset of edges intersecting all triangles in G, that is, the graph becomes triangle-free after deleting these edges. In this chapter, we consider threshold graphs and even-balanced co-chain graphs. For these graph classes, we verify the famous conjecture by Tuza from 1981, Conjecture 1, stating that half of the minimum size of a triangle hitting in G (denoted by $\tau(G)$) is upper bounded the maximum size of a triangle packing in G (denoted by $\mu(G)$).

First, we mention simple properties in Section 7.1 and after that, we prove the conjecture for the two mentioned graph classes separately, in Sections 7.2 and 7.3.

7.1 Preliminaries

Let us first recall the following well-known property (chromatic index of a clique).

Lemma 90. *The edge set of a clique K on k vertices can be decomposed into k edge disjoint maximal matchings for k odd and $k-1$ edge disjoint maximal matchings for k even.*

Proof. If k is even, we may identify the vertices of K with the set $\{0, 1, \ldots, k-1\}$ and consider matchings

$$M_i = \{\{0, i\}\} \cup \{\{a, b\} \mid a \neq b,\ ab \neq 0,\ a + b \equiv 2i \pmod{k-1}\}$$

for $1 \leqslant i \leqslant k-1$. These matchings are edge disjoint and cover the entire edge set of K (cf. Figure 7.1). Removing any vertex (along with all incident edges) yields a desired matching decomposition into $k-1$ matchings of the edge set of the clique of $k-1$ vertices. □

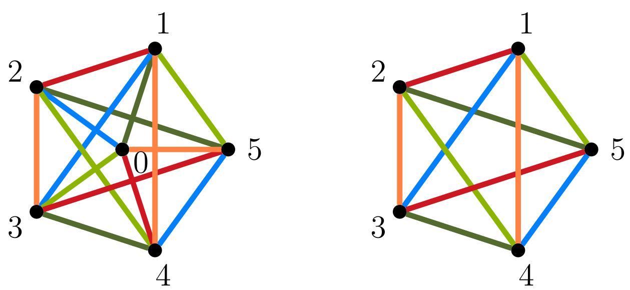


Figure 7.1 The decomposition of edges of a 6-vertex clique into 5 matchings and the corresponding decomposition of a 5-vertex clique.

A graph G is a *complete split graph* if its vertex set can be partitioned into sets K and S, such that S is independent, K induces a clique, and K and S are complete to each other.

The following lemma describes how to pack triangles in complete split graphs. As it is very central to our proofs later, we include a proof here.

Lemma 91 ([87]). *Let K be a clique, S an independent set such that they are complete to each other and $|K| = |S| = k$. Then we can find an (optimal) triangle packing* TP *of size $\binom{k}{2}$ such that:*

1. *It uses all edges from K and each triangle in* TP *contains exactly one edge from K.*

2. *If k is odd, the remaining edges (not used in* TP*) create a matching between K and S, otherwise they create a star with its center vertex in S. Moreover, we can choose the unused matching and the center vertex of the unused star arbitrarily.*

Proof. Consider a graph G composed of a clique K' complete to an independent set S' with $|K'| = k$ and $|S'| = k - 1$, where k is even. By Lemma 90, K can be decomposed into $k - 1$ edge disjoint (perfect) matchings of size $k/2$. Each such matching fully joined to a different vertex in S' yields a family of $k/2$ edge disjoint triangles (see Figure 7.2). The collection of all $k - 1$ such joins is a decomposition of the entire edge set of G into triangles.

Removing any vertex u from K' yields a balanced graph with both sides of odd size, in which edges not packed into triangles (participating in triangles whose vertex u got removed) create a matching between $K' - u$ and S'. On the other hand, by adding a single vertex v to S', we get a balanced graph with both sides of even size, in which unpacked edges form a star (with v being its center vertex). □

Figure 7.2 Full joins of matchings in K with vertices in S as families of triangles.

Corollary 92. *Let K be a clique and S an independent set such that they are complete to each other.*

(a) If $|S| < |K|$, then we can find a triangle packing of size $|S| \cdot \lfloor |K|/2 \rfloor$.

(b) If $|S| \geqslant |K|$, then we can find a triangle packing of size $\binom{|K|}{2}$.

Proof. If $|S| < |K|$, we take arbitrary $|S|$ edge-disjoint maximal matchings in K whose existence follows from Lemma 90 and assign them to different vertices in S. The full join of each such pair consists of $\lfloor |K|/2 \rfloor$ edge-disjoint triangles.

If $|S| \geqslant |K|$, we can derive the statement from Lemma 91: it is enough to take any $|K|$-element subset S' of S. □

We say that we *pack edges of K with vertices of S* when we use triangle packings from Corollary 92. The following lemma describes tightly how many edge-disjoint triangles can be packed in a clique.

Lemma 93 ([87]). *The optimal triangle packing for K_n with $n = 6x + i, 0 \leq i \leq 5$ is $\left(\binom{n}{2} - k\right)/3$ where k is the number of not covered edges and*

- $k = 0$ *for* $i = 1, 3$,
- $k = 4$ *for* $i = 5$,
- $k = \frac{n}{2}$ *for* $i = 0, 2$,
- $k = \frac{n}{2} + 1$ *for* $i = 4$.

Observe, that we can always hit all the triangles in a clique by leaving a bipartite graph with partitions of as equal size as possible and removing the rest. Therefore, the optimal triangle hitting in a clique consists of at most half the edges.

7.2 Threshold Graphs

We recall that a graph $G = (V, E)$ is a *threshold graph* if its vertex set can be partitioned into two sets $K = \{c_1, \ldots, c_k\}$ and $S = \{u_1, \ldots, u_s\}$ such that $G[K]$ is a clique and $G[S]$ is an independent set in G, and $N[c_{i+1}] \subseteq N[c_i]$ for all $1 \leqslant i < k$ and $N(u_i) \subseteq N(u_{i+1})$ for all $1 \leqslant i < s$. We identify K with the clique $G[K]$ and say $G = (K \cup S, E)$ is a threshold graph with given *threshold representation* (K, S).

The threshold representation of a threshold graph may not be unique. We prove that it can be chosen such that the clique contains a vertex which is not adjacent to any vertex of the independent set.

Lemma 94. *For every threshold graph $G = (V, E)$ there exists a threshold representation (K, S) such that there is a vertex $v \in K$ with $N(v) \cap S = \emptyset$.*

Proof. We fix a threshold representation (K, S) of G. Suppose for all $v \in K$ holds $N(v) \cap S \neq \emptyset$. Then, since G is a threshold graph, there is a vertex $w \in S$ such that $N(w) = K$. We obtain a new threshold representation (K', S') of G with $K' := K \cup \{w\}$ and $S' := S \setminus \{w\}$. Since S is an independent set, w has no neighbours in S'. □

We can now prove that Conjecture 1 holds for all threshold graphs.

Theorem 9. *If G is a threshold graph, then $\tau(G) \leqslant 2\mu(G)$.*

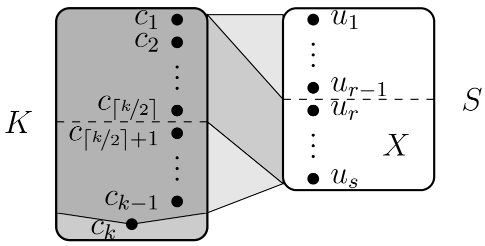
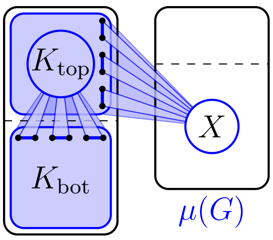
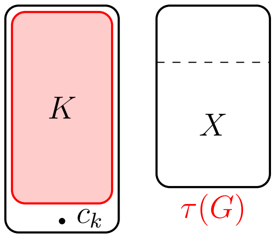


Figure 7.3 The structure of threshold graph G.



(a) triangle packing



(b) triangle hitting

Figure 7.4 The (a) triangle packing and (b) triangle hitting providing the bounds for the case of $|X| \geqslant k/2$.

Proof. Let $G = (K \cup S, E)$ be a threshold graph with $K = \{c_1, \ldots, c_k\}$ and $S = \{u_1, \ldots, u_s\}$ such that $N(c_k) \cap S = \emptyset$. By Lemma 94, such a representation exists. Let $r \in \{1, \ldots, s\}$ be chosen minimal such that $\{c_1, \ldots, c_{\lceil k/2 \rceil}\} \subseteq N(u_r)$ and let X be the subset $\{u_r, \ldots, u_s\}$ of S (see Figure 7.3). Note that X is complete to the set $\{c_1, \ldots, c_{\lceil k/2 \rceil}\}$. We distinguish two cases, based on the parity of k. First, we focus on the case that k is even. In this case we consider two cliques K_{top} and K_{bot} of equal size, induced by vertices $\{c_1, \ldots, c_{k/2}\}$ and $\{c_{k/2+1}, \ldots, c_k\}$, respectively.

We construct a triangle packing TP of G using Corollary 92 as follows: we pack the edges of K_{bot} with vertices in K_{top}, and the edges of K_{top} with vertices in X (see Figure 7.4(a)).

If $|X| \geqslant \frac{k}{2}$, then TP is a triangle packing of size $2\binom{k/2}{2}$. On the other hand, a triangle hitting of size $\binom{k-1}{2}$ can be obtained by taking all edges from K except those incident to c_k (see Figure 7.4(b)). Thus, we obtain a lower bound on the triangle packing and an upper bound on the triangle hitting yielding:

$$\tau(G) \leqslant \binom{k-1}{2} = \frac{k-2}{2} \cdot (k-1) \leqslant \frac{k-2}{2} \cdot k = 4\binom{k/2}{2} \leqslant 2\mu(G).$$

If $|X| < \frac{k}{2}$, then TP is of size at least

$$\binom{k/2}{2} + |X| \cdot \left\lfloor \frac{k}{4} \right\rfloor \geqslant \binom{k/2}{2} + |X| \left(\frac{k}{4} - \frac{1}{2} \right).$$

On the other hand, the edges inside K_{top} and inside K_{bot} together with all edges between S and K_{bot} build a triangle hitting of G (cf. Figure 7.5(b)) of size at most

$$2\binom{k/2}{2} + |X|\left(\frac{k}{2} - 1\right).$$

Indeed, recall that c_k does not have any neighbours in S, therefore we have at most $|X|\left(\frac{k}{2} - 1\right)$ edges between X and K_{bot}, and by definition of X, there are no vertices in K_{bot} having neighbours in $S \setminus X$. Thus, we again obtain a lower bound on the triangle packing and an upper bound on the triangle hitting yielding:

$$\tau(G) \leqslant 2\binom{k/2}{2} + |X|\left(\frac{k}{2} - 1\right) = 2\binom{k/2}{2} + 2\,|X|\left(\frac{k}{4} - \frac{1}{2}\right) \leqslant 2\mu(G).$$

We are left with the case that k is odd. We consider the cliques K_{top} and K_{bot} induced

(a) triangle packing (b) triangle hitting

Figure 7.5 The (a) triangle packing and (b) triangle hitting providing the bounds when $|X| < k/2$.

by sets $\{c_1, \ldots, c_{(k+1)/2}\}$ and $\{c_{(k+1)/2+1}, \ldots, c_k\}$, respectively.

Again, we look at the size of X and in case it is large, we can derive a similar argument as in the previous case, using Corollary 92. More precisely, assume that $|X| \geqslant \frac{k+1}{2}$. Then we pack the edges of K_{bot} into $\binom{(k-1)/2}{2}$ triangles with vertices in K_{top}, and the edges of K_{top} into $\binom{(k+1)/2}{2}$ triangles with vertices in X. Together, this gives a triangle packing of size

$$\binom{\frac{k+1}{2}}{2} + \binom{\frac{k-1}{2}}{2} = \frac{(k-1)^2}{4}.$$

The triangle hitting again consists of all edges from K except those adjacent to c_k, therefore has size $\binom{k-1}{2}$ (recall Figure 7.4). These two bounds together yield:

$$\tau(G) \leqslant \binom{k-1}{2} = \frac{k-1}{2} \cdot (k-2) \leqslant \frac{(k-1)^2}{2} \leqslant 2\mu(G).$$

It remains to consider the case $|X| < \frac{k+1}{2}$. In order to find a triangle packing, we define K'_{top} and K'_{bot} to be induced by $\{c_1, \ldots, c_{(k-1)/2}\}$ and $\{c_{(k+1)/2}, \ldots, c_k\}$, respectively (so

$K'_{\text{top}} = K_{\text{top}} \setminus \{c_{(k+1)/2}\}$ is of size $\frac{k-1}{2}$ and $K'_{\text{bot}} = K_{\text{bot}} \cup \{c_{(k+1)/2}\}$ is of size $\frac{k+1}{2}$). We build a triangle packing analogously to before, using Corollary 92. The edges of K'_{bot} can be packed into $\lfloor \frac{(k+1)/2}{2} \rfloor \cdot \frac{k-1}{2} \geqslant \frac{k-1}{4} \cdot \frac{k-1}{2}$ triangles with vertices in K'_{top}. Moreover, $\min\{|X| \cdot \lfloor \frac{k-1}{4} \rfloor, \binom{(k-1)/2}{2}\} \geqslant |X| \frac{k-3}{4}$ edges of K'_{top} can be packed into triangles with vertices in X (see Figure 7.6(a)). This gives a triangle packing of size at least

$$\frac{k-1}{2} \cdot \frac{k-1}{4} + |X| \frac{k-3}{4}.$$

To find a triangle hitting, we again consider the partition of K into K_{top} and K_{bot}. We take all edges inside K_{top} and inside K_{bot} together with all edges between S and K_{bot} (see Figure 7.6(b)). Again, recall that $c_k \in K_{\text{bot}}$ does not have any neighbours in S, and there are no vertices in K_{bot} having neighbours in $S \setminus X$. Thus, this yields a triangle hitting of size at most.

$$\binom{\frac{k+1}{2}}{2} + \binom{\frac{k-1}{2}}{2} + |X| \frac{k-3}{2}.$$

Therefore, we obtain the following which concludes the proof:

$$\begin{aligned} \tau(G) &\leqslant \binom{\frac{k+1}{2}}{2} + \binom{\frac{k-1}{2}}{2} + |X| \frac{k-3}{2} \\ &= \frac{(k-1)^2}{4} + |X| \frac{k-3}{2} = 2 \cdot \frac{k-1}{2} \cdot \frac{k-1}{4} + 2\,|X| \frac{k-3}{4} \leqslant 2\mu(G). \qquad \square \end{aligned}$$

(a) triangle packing (b) triangle hitting

Figure 7.6 In (a) the triangle packing and in (b) the triangle hitting providing the bounds for $|K|$ odd and $|X| < (k+1)/2$.

7.3 Even Balanced Co-chain Graphs

In this section we prove Theorem 10. Let us restate it here.

Theorem 10. *If G is an even balanced co-chain graph, then $\tau(G) \leqslant 2\mu(G)$.*

To this end let G be an even balanced co-chain graph and (K_1, K_2) its *co-chain representation*. Recall that K_1 and K_2 are of the same size which is divisible by 4, for the rest of the section let $|K_1| = |K_2| = 2\ell$ for ℓ even. We identify K_1 and K_2 with the cliques $G[K_1]$ and $G[K_2]$. See Figure 7.7 for an illustration.

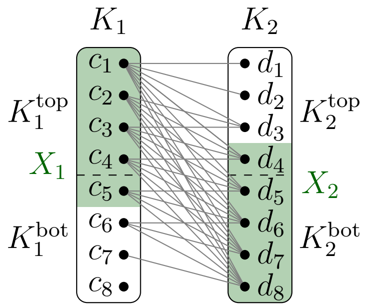


Figure 7.7 An example of an even balanced co-chain graph with $\ell = 4$ (omitting the edges inside the cliques K_1 and K_2).

Proof of Theorem 10. Note that in the case $\ell = 2$ we get an 8-vertex graph which is either a clique, or has average degree less than 7, so this case is covered by [161]. Therefore in the following we assume that $\ell \geqslant 4$.

Similarly to threshold graphs, we use $K_1^{\text{top}}, K_1^{\text{bot}}$ for the top and the bottom half of K_1, respectively, and similarly $K_2^{\text{top}}, K_2^{\text{bot}}$ for the top and the bottom half of K_2. Let $X_1 \subseteq K_1, X_2 \subseteq K_2$ be the sets defined as follows: $c \in X_1$ if $K_2^{\text{bot}} \subseteq N[c]$, and $d \in X_2$ if $K_1^{\text{top}} \subseteq N[d]$. See Figure 7.7 for an illustration. We denote $x_1 = |X_1|$ and $x_2 = |X_2|$. By definition, $x_1 \geq \ell$ implies that the set $X_1 \supseteq K_1^{\text{top}}$ is complete to K_2^{bot}. Consequently, $x_2 \geq \ell$. Similarly, $x_2 \geq \ell$ implies $x_1 \geq \ell$. Therefore, $x_1 \geq \ell$ if and only if $x_2 \geq \ell$. We assume without loss of generality throughout the entire proof that $x_1 \geqslant x_2$. We split the analysis into two main cases.

7.3.1 The Case $x_1, x_2 \leq \ell$

In this case $X_1 \subseteq K_1^{\text{top}}$ and $X_2 \subseteq K_2^{\text{bot}}$. Suppose there is an edge cd with $c \in K_1 \setminus X_1$ and $d \in K_2^{\text{top}}$, then c is adjacent to all the vertices in K_2^{bot} and so $c \in X_1$, which yields a contradiction. Similarly, there are no edges between K_1^{bot} and $K_2 \setminus X_2$. In particular, there are no edges between K_2^{top} and K_1^{bot}.

We choose a triangle hitting TH obtained by taking all edges within K_1^{top}, K_2^{top}, K_1^{bot}, and K_2^{bot}, as well as edges between X_1 and K_2^{bot}, and between X_2 and K_1^{top} as illustrated in Figure 7.8. Observe now that in the graph $G - \mathsf{TH}$ vertices in X_1 only have neighbours in the independent set $K_1^{\text{bot}} \cup K_2^{\text{top}}$, vertices in $K_1^{\text{top}} \setminus X_1$ only have neighbours in the independent set $K_1^{\text{bot}} \cup K_2^{\text{bot}} \setminus X_2$, while vertices in K_1^{bot} only have neighbours in the independent set $K_1^{\text{top}} \cup X_2$. Therefore the set TH is indeed a triangle hitting of G.

Therefore,

$$\tau(G) \leq |\mathsf{TH}| = 4\binom{\ell}{2} + \ell x_1 + \ell x_2 - x_1 x_2 = 4\binom{\ell}{2} + \ell x_1 + (\ell - x_1)x_2.$$

Indeed, we note that we counted edges between X_1 and X_2 once in term ℓx_1 and once in term ℓx_2 which we compensate by subtracting the last term $x_1 x_2$.

Let us now create a sufficiently large triangle packing. First, we pack all edges of K_1^{bot} with vertices in K_1^{top} and also all edges of K_2^{top} with vertices in K_2^{bot}; we denote the set of these triangles by A (see Figure 7.9(a)). By Lemma 91, A contains $2\binom{\ell}{2}$ triangles. Observe that $2|A| - |\mathsf{TH}| = -\ell x_1 - (\ell - x_1)x_2$. First, we sort out the single case where $x_1 = \ell$,

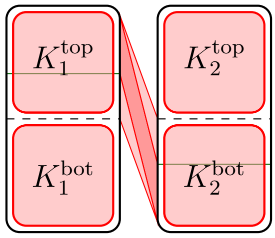
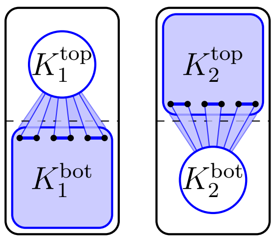


Figure 7.8 The triangle hitting used in the case $x_1, x_2 \leqslant \ell$.

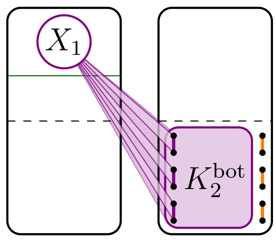
and, in consequence, $x_2 = \ell$ by definition of X_1 and X_2 together with the assumption that $x_2 \leq \ell$.

The Subcase $x_1 = x_2 = \ell$

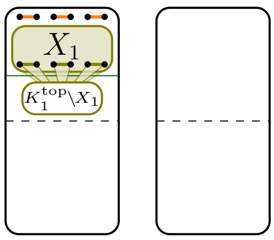
In this case, $|\mathsf{TH}| = 4\binom{\ell}{2} + \ell^2$. As $K_1^{\text{top}} \cup K_2^{\text{bot}}$ is a clique, by Lemma 93 we can pack at least $\frac{1}{3}\left(\binom{2\ell}{2} - \ell - 1\right)$ triangles in it. Together with A, we obtain a triangle packing TP. If $\ell \geqslant 5$, then $2\mathsf{TP} - \mathsf{TH} \geq \frac{2}{3}\left(\binom{2\ell}{2} - \ell - 1\right) - \ell^2 = \frac{1}{3}\left(\ell^2 - 4\ell - 2\right) \geq 0$. If $\ell = 4$, Lemma 93 gives us a stronger bound without the term -2, leading to $2\mathsf{TP} - \mathsf{TH} \geq \frac{1}{3}\left(\ell^2 - 4\ell\right) = 0$. Both cases imply $2\mu(G) \geqslant \tau(G)$.



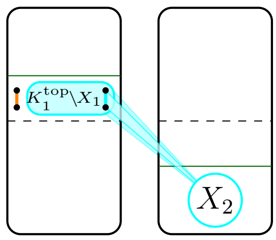
(a) set A of triangles



(b) set B of triangles



(c) set C of triangles



(d) set D of triangles

Figure 7.9 Triangles in (a) A, (b) B, (c) C, and (d) D in the case $x_1, x_2 \leqslant \ell$.

The Subcase $x_1, x_2 < \ell$

Now, we consider the remaining case where $x_1 < \ell$, and, in consequence, $x_2 < \ell$.

We choose a triangle packing TP as follows (see Figure 7.9). We take the set A of triangles as defined before. Recall that $2|A| - |\mathsf{TH}| = -\ell x_1 - (\ell - x_1)x_2$. We create a set B of triangles by packing edges of K_2^{bot} with vertices in X_1. By Corollary 92(a) and as $x_1 < \ell$, B is of size $\ell/2 \cdot x_1$. We create another set of triangles C by packing edges of X_1 with vertices of $K_1^{\text{top}} \setminus X_1$. Next, let D be the set of triangles created by packing edges of $K_1^{\text{top}} \setminus X_1$ with vertices in X_2. It is clear that all triangles in $\mathsf{TP} = A \cup B \cup C \cup D$ are mutually edge-disjoint, therefore TP is indeed a triangle packing.

Let us first settle the case that x_1 is even. As $2(|A| + |B|) - |\mathsf{TH}| = -(\ell - x_1)x_2$ if $x_1 < \ell$, it remains to show that $2\,|\mathsf{TP} \setminus (A \cup B)| = 2(|C| + |D|) \geqslant (\ell - x_1)x_2$.

If $\ell - x_1 > x_2$, then $2|D| = (\ell - x_1)x_2$ by Corollary 92(a). So, assume that $\ell - x_1 \leq x_2$. Consequently, $\ell - x_1 \leq x_1$ and thus $\ell/2 \leqslant x_1$. If $x_1 = \ell/2$, then, by $x_1 \geq x_2 \geq \ell/2$, we have $x_2 = \ell/2$ as well. Thus, as $\ell \geqslant 4$, $2(|C| + |D|) - (\ell - x_1)x_2 = 4\binom{\ell/2}{2} - \ell^2/4 = \ell(\ell-4)/4 \geqslant 0$. For $\ell - x_1 < x_1$ we get $2|C| = x_1(\ell - x_1) \geq x_2(\ell - x_1)$. Therefore, we always have $2\,|C \cup D| \geqslant (\ell - x_1)x_2$ for even x_1, and so $2\mu(G) \geq 2\mathsf{TP} \geq \mathsf{TH} \geq \tau(G)$.

In case x_1 is odd, we add one additional triangle to our triangle packing as follows. Note that if there is no edge between K_1^{bot} and K_2^{bot}, then all edges between K_1^{top} and K_2^{top} hit all triangles between K_1 and K_2, therefore taking these edges instead of edges between K_1^{top} and K_2^{bot} creates a triangle hitting TH' of size at most $4\binom{\ell}{2} + x_1\ell$ as all the edges between K_1^{top} and K_2^{top} have one endpoint in X_1. As $x_1 < \ell$, we obtain $2\mu(G) \geqslant 2(|A| + |B|) \geqslant |\mathsf{TH}'| \geqslant \tau(G)$. Thus we can assume that there is at least one edge uv with $u \in K_1^{\text{bot}}$ and $v \in K_2^{\text{bot}}$.

Note in particular that $v \in X_2$ as every edge between K_1^{bot} and K_2^{bot} has one endpoint in X_2. Observe that $|K_1^{\text{top}} \setminus X_1| = \ell - x_1$ is odd, so there exists an unpacked matching between $K_1^{\text{top}} \setminus X_1$ and X_2 (not containing edges used in triangles from set D). Indeed, each maximal matching in $K_1^{\text{top}} \setminus X_1$ constructed according to Lemma 90 omits a different vertex $u_1 \in K_1^{\text{top}} \setminus X_1$, so after the matching is fully joined with a vertex $u_2 \in X_2$, as in Corollary 92, the edge u_1u_2 remains unpacked. A collection of all such edges gives the desired matching. Let $w \in K_1^{\text{top}} \setminus X_1$ be a vertex such that wv is an edge of the mentioned unpacked matching. Finally, as ℓ is even, a star with center in K_1^{top} is not used in any triangle in A, by Lemma 91. Note that the center of this star can be chosen arbitrarily among vertices of K_1^{top} by Lemma 91; let us choose w to be the center. Therefore, uvw is a triangle which is edge-disjoint with every triangle in $A \cup B \cup C \cup D$ and we may set $\mathsf{TP}^{\text{odd}} = \mathsf{TP} \cup \{uvw\}$ for odd x_1.

Recall that $2(|A| + |B|) - |\mathsf{TH}| = -(\ell - x_1)x_2$. Similarly as before, we need to prove that

$$2\left|\mathsf{TP}^{\text{odd}} \setminus (A \cup B)\right| = 2(|C| + |D| + 1) \geqslant (\ell - x_1)x_2.$$

If $\ell - x_1 \leq x_2$, then again $\ell - x_1 \leq x_1$ and thus $\ell/2 \leqslant x_1$. The case $\ell/2 = x_1$ can be handled exactly as in the even case. So assume further $\ell - x_1 < x_1$, then using Corollary 92 we obtain $2(|C| + |D|) = (x_1 - 1)(\ell - x_1) + 2\binom{\ell - x_1}{2} = (x_1 - 1)(\ell - x_1) + (\ell - x_1)(\ell - x_1 - 1) = (\ell - x_1)(\ell - 2)$. Consequently, $2(|C| + |D| + 1) - (\ell - x_1)x_2 = 2 + (\ell - x_1)(\ell - 2 - x_2)$. Observe that, for $x_2 \leq \ell - 2$, we already get $(\ell - x_1)(\ell - 2 - x_2) \geq 0$. We have $x_1 = \ell - 1$ because x_1 is odd and ℓ is even. For $x_2 = \ell - 1$, we have $x_1 = \ell - 1$

because $x_2 \leq x_1 < \ell$. Thus $2 + (\ell - x_1)(\ell - 2 - x_2) = 2 + 1 \cdot (-1) \geqslant 0$. Therefore, we obtain the inequality $2\left(|C| + |D| + 1\right) \geqslant (\ell - x_1)x_2$.

If $\ell - x_1 > x_2$, then $2|D| = (\ell - x_1 - 1)x_2 = (\ell - x_1)x_2 - x_2$. Hence in this case, D alone does not suffice as it is missing x_2 triangles. We therefore need $2\,|C| + 2 \geqslant x_2$. We use Corollary 92 to analyse the size of C.

If $x_1 \leq \ell - x_1$, then $2|C| + 2 - x_2 \geq x_1(x_1 - 1) - x_2 + 2 \geq (x_2 - 1)^2 + 1 \geq 1$ as $x_1(x_1-1) \geq x_2(x_2-1)$. If $x_1 > \ell - x_1$, then, $2|C| + 2 - x_2 = (x_1 - 1)(\ell - x_1) - x_2 + 2 \geq x_1 - x_2 + 1 \geqslant 1$, as $\ell - x_1 \geqslant 1$ and $x_1 \geq x_2$. So in both cases we obtain $2\,|C| + 2 \geqslant x_2 + 1 \geqslant x_2$.

We conclude that $2\mu(G) \geq 2\mathsf{TP}^{\text{odd}} \geq \mathsf{TH} \geq \tau(G)$.

7.3.2 The Case $x_1 > \ell$ and $x_2 \geq \ell$

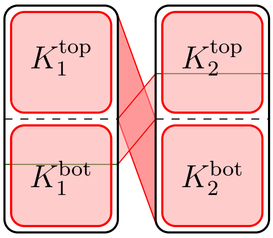


Figure 7.10 The triangle hitting used in the case $x_1 > \ell$ and $x_2 \geq \ell$.

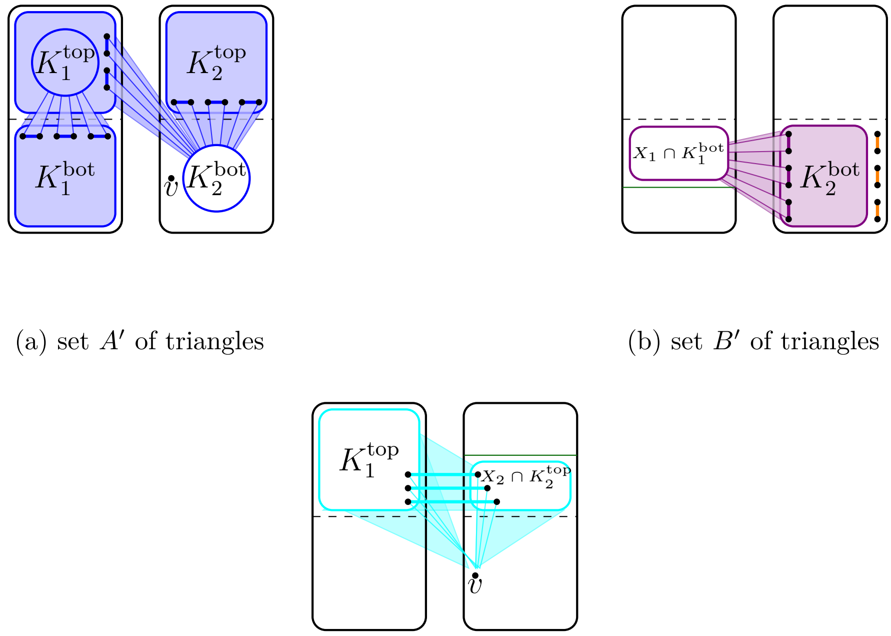
We choose a triangle hitting TH obtained by taking all edges within K_1^{top}, K_1^{bot}, K_2^{top} and K_2^{bot} as well as edges between K_1^{top} and K_2^{bot} and between K_1^{bot} and K_2^{top} (cf. Figure 7.10). The graph $G - \mathsf{TH}$ is bipartite, thus TH is indeed a triangle hitting in G. We have

$$|\mathsf{TH}| = 4\binom{\ell}{2} + \ell^2 + \left|E\left(K_2^{\text{top}}, K_1^{\text{bot}}\right)\right| \leqslant 3\ell^2 - 2\ell + (x_1 - \ell)(x_2 - \ell).$$

We choose a triangle packing TP as follows. Pack all edges of K_2^{top} with vertices of K_2^{bot}, all edges of K_1^{top} with vertices in K_2^{bot} and all edges of K_1^{bot} with vertices in K_1^{top}. This gives a set A' of $3\binom{\ell}{2}$ triangles (see Figure 7.11(a)). By the second part of Lemma 91 there exists $v \in K_2^{\text{bot}}$ such that edges between v and $K_2^{\text{top}} \cup K_1^{\text{top}}$ are not used in A'. Additionally, define a set B' of triangles obtained by packing edges from K_2^{bot} with vertices of $X_1 \cap K_1^{\text{bot}}$ (see Figure 7.11(b)). Then $|B'| = \frac{\ell}{2}(x_1 - \ell)$ if $x_1 \neq 2\ell$ and $|B'| = \binom{\ell}{2}$ (by Corollary 92(b)) if $x_1 = 2\ell$. Finally, let C' be the set of triangles using v and any maximal matching between K_1^{top} and $X_2 \cap K_2^{\text{top}}$ (see Figure 7.11(c)). Since K_1^{top} is complete to $X_2 \cap K_2^{\text{top}}$, we obtain $|C'| = x_2 - \ell$. It is clear that $\mathsf{TP} = A' \cup B' \cup C'$ is a triangle packing.

If $x_1 < 2\ell$, then

$$\begin{aligned} 2\,|\mathsf{TP}| - |\mathsf{TH}| &\geqslant 3\ell(\ell - 1) + \ell(x_1 - \ell) + 2(x_2 - \ell) - 3\ell^2 + 2\ell - (x_1 - \ell)\,(x_2 - \ell) \\ &= (x_1 - \ell - 1)\,(2\ell - x_2) + x_2 - \ell \geqslant 0. \end{aligned}$$



(a) set A' of triangles

(b) set B' of triangles

(c) set C' of triangles

Figure 7.11 Triangles in (a) A', (b) B', and (c) C' in the case $x_1 > \ell$ and $x_2 \geq \ell$.

The last inequality follows as $x_1 \geq \ell + 1$.

If $x_1 = 2\ell$, then we similarly get

$$\begin{aligned} 2\,|\mathsf{TP}| - |\mathsf{TH}| &\geqslant 3\ell(\ell-1) + \ell(\ell-1) + 2(x_2 - \ell) - 3\ell^2 + 2\ell - \ell\,(x_2 - \ell) \\ &= (\ell - 2)\,(2\ell - x_2) \geqslant 0. \end{aligned}$$

We conclude that indeed $2\mu(G) \geq \tau(G)$. $\square$

Bibliography

[1] Tara Abrishami, Maria Chudnovsky, Cemil Dibek, and Paweł Rzążewski. Polynomial-time algorithm for maximum independent set in bounded-degree graphs with no long induced claws. *CoRR*, abs/2107.05434, 2021. arXiv:2107.05434.

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

[3] Tara Abrishami, Maria Chudnovsky, Cemil Dibek, Stéphan Thomassé, Nicolas Trotignon, and Kristina Vušković. Graphs with polynomially many minimal separators. *J. Comb. Theory, Ser. B*, 152:248–280, 2022. doi:10.1016/j.jctb.2021.10.003.

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

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

[6] Michael O. Albertson and David M. Berman. A conjecture on planar graphs. *Graph theory and related topics*, 357:1, 1979.

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

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

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

[10] Alberto Rojas Anríquez and Maya Stein. 3-colouring p_t-free graphs without short odd cycles. *Algorithmica*, 85(4):831–853, 2023. doi:10.1007/S00453-022-01049-0.

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

[12] Kirby A Baker, Peter C Fishburn, and Fred S Roberts. Partial orders of dimension 2. *Networks*, 2(1):11–28, 1972.

[13] Jacob D. Baron and Jeff Kahn. Tuza's conjecture is asymptotically tight for dense graphs. *Combinatorics, Probability and Computing*, 25(5):645–667, 2016. doi:10.1017/S0963548316000067.

[14] Alexandre Blanché, Konrad K. Dabrowski, Matthew Johnson, Vadim V. Lozin, Daniël Paulusma, and Viktor Zamaraev. Clique-width for graph classes closed under complementation. *SIAM Journal on Discrete Mathematics*, 34(2):1107–1147, 2020. doi:10.1137/18M1235016.

[15] Hans L. Bodlaender, Nick Brettell, Matthew Johnson, Giacomo Paesani, Daniël Paulusma, and Erik Jan van Leeuwen. Steiner trees for hereditary graph classes: A treewidth perspective. *Theoretical Computer Science*, 10(1016):30–39, 2021. doi:10.1016/j.tcs.2021.03.012.

[16] Rodica Boliac and Vadim V. Lozin. On the clique-width of graphs in hereditary classes. In *Proc. ISAAC 2002, LNCS*, pages 44–54, 2002. doi:10.1007/3-540-36136-7_5.

[17] Marthe Bonamy, Nicolas Bousquet, Konrad K. Dabrowski, Matthew Johnson, Daniël Paulusma, and Théo Pierron. Graph isomorphism for (H_1, H_2)-free graphs: an almost complete dichotomy. *Algorithmica*, 83(3):822–852, 2021. doi:10.1007/s00453-020-00747-x.

[18] Marthe Bonamy, Łukasz Bożyk, Andrzej Grzesik, Meike Hatzel, Tomáš Masařík, Jana Novotná, and Karolina Okrasa. Tuza's conjecture for threshold graphs. In Jaroslav Nešetřil, Guillem Perarnau, Juanjo Rué, and Oriol Serra, editors, *Extended Abstracts EuroComb 2021*, pages 765–771, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-83823-2_122.

[19] Marthe Bonamy, Łukasz Bożyk, Andrzej Grzesik, Meike Hatzel, Tomáš Masařík, Jana Novotná, and Karolina Okrasa. Tuza's conjecture for threshold graphs. *Discrete Mathematics & Theoretical Computer Science*, 24(1), 2022. doi:10.46298/DMTCS.7660.

[20] Flavia Bonomo, Maria Chudnovsky, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-Coloring and List Three-Coloring of Graphs Without Induced Paths on Seven Vertices. *Combinatorica*, 38(4):779–801, May 2017. doi:10.1007/s00493-017-3553-8.

[21] Flavia Bonomo, Oliver Schaudt, and Maya Stein. 3-colouring graphs without triangles or induced paths on seven vertices. *CoRR*, 2014. arXiv:1410.0040v1.

[22] Fábio Botler, Cristina G. Fernandes, and Juan Gutiérrez. On Tuza's conjecture for triangulations and graphs with small treewidth. *Discrete Mathematics*, 344(4):112281, 2021. doi:10.1016/j.disc.2020.112281.

[23] Vincent Bouchitté and Ioan Todinca. Listing all potential maximal cliques of a graph. *Theoretical Computer Science*, 276(1):17–32, 2002. doi:10.1016/S0304-3975(01)00007-X.

[24] Romain Bourneuf, Jana Masaříková, Wojciech Nadara, and Marcin Pilipczuk. Max Weight Independent Set in Graphs with No Long Claws: A Stronger Analog of the Gyárfás' Path Argument, 2024+. manuscript.

[25] Łukasz Bożyk, Jan Derbisz, Tomasz Krawczyk, Jana Novotná, and Karolina Okrasa. Vertex deletion into bipartite permutation graphs. In Yixin Cao and Marcin Pilipczuk, editors, *15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14–18, 2020, Hong Kong, China (Virtual Conference)*, volume 180 of *LIPIcs*, pages 5:1–5:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.IPEC.2020.5.

[26] Łukasz Bożyk, Jan Derbisz, Tomasz Krawczyk, Jana Novotná, and Karolina Okrasa. Vertex deletion into bipartite permutation graphs. *Algorithmica*, 84(8):2271–2291, 2022. doi:10.1007/S00453-021-00923-7.

[27] Peter Bradshaw, Tomáš Masařík, Jana Novotná, and Ladislav Stacho. Robust connectivity of graphs on surfaces. *SIAM Journal on Discrete Mathematics*, 36(2):1416–1435, June 2022. doi:10.1137/21m1417077.

[28] Peter Bradshaw, Tomáš Masařík, and Ladislav Stacho. Flexible list colorings in graphs with special degeneracy conditions. *Journal of Graph Theory*, 101(4):717–745, 2022. doi:10.1002/JGT.22849.

[29] Andreas Brandstädt, Konrad K. Dabrowski, Shenwei Huang, and Daniël Paulusma. Bounding the clique-width of H-free split graphs. *Discrete Applied Mathematics*, 10(1016):30–39, 2016. doi:10.1016/j.dam.2016.04.003.

[30] Andreas Brandstädt, Konrad K. Dabrowski, Shenwei Huang, and Daniël Paulusma. Bounding the clique-width of H-free chordal graphs. *Journal of Graph Theory*, 86(1):42–77, 2017. doi:10.1002/jgt.22111.

[31] Andreas Brandstädt, Joost Engelfriet, Hoàng-Oanh Le, and Vadim V. Lozin. Clique-width for 4-vertex forbidden subgraphs. *Theory of Computing Systems*, 39(4):561–590, 2006. doi:10.1007/s00224-005-1199-1.

[32] Andreas Brandstädt and Chính T. Hoàng. On clique separators, nearly chordal graphs, and the maximum weight stable set problem. *Theoretical Computer Science*, 389(1-2):295–306, 2007. doi:10.1016/j.tcs.2007.09.031.

[33] Andreas Brandstädt, Tilo Klembt, and Suhail Mahfud. P_6- and triangle-free graphs revisited: structure and bounded clique-width. *Discrete Mathematics and Theoretical Computer Science*, 8(1):173–188, 2006. doi:10.46298/dmtcs.372.

[34] Andreas Brandstädt and Dieter Kratsch. On the restriction of some NP-complete graph problems to permutation graphs. In *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985*, volume 199 of *Lecture Notes in Computer Science*, pages 53–62. Springer, 1985. doi:10.1007/BFb0028791.

[35] Andreas Brandstädt, Hoàng-Oanh Le, and Raffaele Mosca. Gem- and co-gem-free graphs have bounded clique-width. *International Journal of Foundations of Computer Science*, 15(1):163–185, 2004. doi:10.1142/S0129054104002364.

[36] Andreas Brandstädt, Hoàng-Oanh Le, and Raffaele Mosca. Chordal co-gem-free and (P_5,gem)-free graphs have bounded clique-width. *Discrete Applied Mathematics*, 145(2):232–241, 2005. doi:10.1016/j.dam.2004.01.014.

[37] Andreas Brandstädt and Suhail Mahfud. Maximum weight stable set on graphs without claw and co-claw (and similar graph classes) can be solved in linear time. *Information Processing Letters*, 84(5):251–259, 2002. doi:10.1016/S0020-0190(02)00291-0.

[38] Andreas Brandstädt and Raffaele Mosca. On distance-3 matchings and induced matchings. *Discrete Applied Mathematics*, 159(7):509–520, 2011. doi:10.1016/j.dam.2010.05.022.

[39] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.

[40] Kathie Cameron, Murilo Vicente Gonçalves da Silva, Shenwei Huang, and Kristina Vušković. Structure and algorithms for (cap, even hole)-free graphs. *Discrete Mathematics*, 341(2):463–473, 2018. doi:10.1016/j.disc.2017.09.013.

[41] Wouter Cames van Batenburg, Tony Huynh, Gwenaël Joret, and Jean-Florent Raymond. A tight Erdős-Pósa function for planar minors. *Advances in Combinatorics*, 2019:33, 2019. Id/No 2. doi:10.19086/aic.10807.

[42] Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1096–1115. SIAM, 2016. doi:10.1137/1.9781611974331.ch77.

[43] Yixin Cao. Unit interval editing is fixed-parameter tractable. *Information and Computation*, 253:109–126, 2017. doi:10.1016/j.ic.2017.01.008.

[44] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016. doi:10.1007/s00453-015-0014-x.

[45] Parinya Chalermsook, Samir Khuller, Pattara Sukprasert, and Sumedha Uniyal. Multi-transversals for triangles and the Tuza's conjecture. In *Proceedings of SODA 2020*, pages 1955–1974. SIAM, 2020. doi:10.5555/3381089.3381210.

[46] Claude A Christen and Stanley M Selkow. Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B*, 27(1):49–59, August 1979. doi:10.1016/0095-8956(79)90067-4.

[47] Maria Chudnovsky, Shenwei Huang, Sophie Spirkl, and Mingxian Zhong. List 3-Coloring Graphs with No Induced $P_6 + rP_3$. *Algorithmica*, July 2020. doi:10.1007/s00453-020-00754-y.

[48] Maria Chudnovsky, Peter Maceli, Juraj Stacho, and Mingxian Zhong. 4-Coloring P_6-Free Graphs with No Induced 5-Cycles. *Journal of Graph Theory*, 84(3):262–285, 2017. doi:10.1002/jgt.22025.

[49] Maria Chudnovsky, Peter Maceli, and Mingxian Zhong. Three-coloring graphs with no induced seven-vertex path I : the triangle-free case. *CoRR*, 2014. arXiv:1409.5164.

[50] Maria Chudnovsky, Peter Maceli, and Mingxian Zhong. Three-coloring graphs with no induced seven-vertex path II : using a triangle. *CoRR*, 2015. arXiv:1503.03573.

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

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

[53] Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs. *SIAM Journal on Computing*, 53(1):47–86, 2024. doi:10.1137/20M1333778.

[54] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, 2006. doi:10.4007/annals.2006.164.51.

[55] Maria Chudnovsky and Paul D. Seymour. The structure of claw-free graphs. In Bridget S. Webb, editor, *Surveys in Combinatorics, 2005 [invited lectures from the Twentieth British Combinatorial Conference, Durham, UK, July 2005]*, volume 327 of *London Mathematical Society Lecture Note Series*, pages 153–171. Cambridge University Press, 2005. doi:10.1017/cbo9780511734885.008.

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

[57] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. List-three-coloring P_t-free graphs with no induced 1-subdivision of $K_{1,s}$. *Discrete Mathematics*, 343(11):112086, November 2020. doi:10.1016/j.disc.2020.112086.

[58] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-Coloring P_6-Free Graphs. I. Extending an Excellent Precoloring. *SIAM Journal on Computing*, 53(1):111–145, 2024. doi:10.1137/18M1234837.

[59] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-Coloring P_6-Free Graphs. II. Finding an Excellent Precoloring. *SIAM Journal on Computing*, 53(1):146–187, 2024. doi:10.1137/18M1234849.

[60] Maria Chudnovsky and Juraj Stacho. 3-colorable subclasses of P_8-free graphs. *SIAM Journal on Discrete Mathematics*, 32(2):1111–1138, January 2018. doi:10.1137/16m1104858.

[61] Julia Chuzhoy. Improved bounds for the flat wall theorem. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2015, page 256–275, USA, 2015. Society for Industrial and Applied Mathematics. doi:10.5555/2722129.2722149.

[62] D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985. doi:10.1137/0214065.

[63] Derek G. Corneil and Yehoshua Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, September 1984. doi:10.1016/0166-218x(84)90088-x.

[64] Bruno Courcelle and Joost Engelfriet. Graph structure and monadic second-order logic: A language-theoretic approach. In volume 138 of, editor, *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2012. doi:10.1017/CBO9780511977619.

[65] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

[66] Jean-François Couturier, Petr A. Golovach, Dieter Kratsch, and Daniël Paulusma. List coloring in the absence of a linear forest. *Algorithmica*, 71(1):21–35, April 2013. doi:10.1007/s00453-013-9777-0.

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

[68] Konrad K. Dabrowski, François Dross, and Daniël Paulusma. Colouring diamond-free graphs. *Journal of Computer and System Sciences*, 10(1016):410–431, 2017. doi:10.1016/j.jcss.2017.06.005.

[69] Konrad K. Dabrowski, Petr A. Golovach, and Daniël Paulusma. Colouring of graphs with Ramsey-type forbidden subgraphs. *Theoretical Computer Science*, 10(1016):34–43, 2014. doi:10.1016/j.tcs.2013.12.004.

[70] Konrad K. Dabrowski, Shenwei Huang, and Daniël Paulusma. Bounding clique-width via perfect graphs. *Journal of Computer and System Sciences*, 10(1016):202–215, 2019. doi:10.1016/j.jcss.2016.06.007.

[71] Konrad K. Dabrowski, Matthew Johnson, and Daniël Paulusma. Clique-width for hereditary graph classes. *London Mathematical Society Lecture Note Series*, 10(1017):1–56, 2019. doi:10.1017/9781108649094.002.

[72] Konrad K. Dabrowski, Vadim V. Lozin, and Daniël Paulusma. Clique-width and well-quasi-ordering of triangle-free graph classes. *Journal of Computer and System Sciences*, 10(1016):64–91, 2020. doi:10.1016/j.jcss.2019.09.001.

[73] Konrad K. Dabrowski, Vadim V. Lozin, Rajiv Raman, and Bernard Ries. Colouring vertices of triangle-free graphs without forests. *Discrete Mathematics*, 312(7):1372–1385, 2012. doi:10.1016/j.disc.2011.12.012.

[74] Konrad K. Dabrowski, Tomáš Masařík, Jana Novotná, Daniël Paulusma, and Paweł Rzążewski. Clique-width: Harnessing the power of atoms. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science - 46th International Workshop, WG 2020, Leeds, UK, June 24–26, 2020, Revised Selected Papers*, volume 12301 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 2020. doi:10.1007/978-3-030-60440-0_10.

[75] Konrad K. Dabrowski, Tomáš Masařík, Jana Novotná, Daniël Paulusma, and Paweł Rzążewski. Clique-width: Harnessing the power of atoms. *Journal of Graph Theory*, 104(4):769–810, 2023. doi:10.1002/JGT.23000.

[76] Konrad K. Dabrowski and Daniël Paulusma. Classifying the clique-width of H-free bipartite graphs. *Discrete Applied Mathematics*, 10(1016):43–51, 2016. doi:10.1016/j.dam.2015.06.030.

[77] Konrad K. Dabrowski and Daniël Paulusma. Clique-width of graph classes defined by two forbidden induced subgraphs. *The Computer Journal*, 59(5):650–666, 2016. doi:10.1093/comjnl/bxv096.

[78] Jitender S. Deogun and George Steiner. Polynomial algorithms for hamiltonian cycle in cocomparability graphs. *SIAM Journal on Computing*, 23(3):520–552, 1994. doi:10.1137/S0097539791200375.

[79] Jan Derbisz, Lawqueen Kanesh, Jayakrishnan Madathil, Abhishek Sahu, Saket Saurabh, and Shaily Verma. A polynomial kernel for bipartite permutation vertex deletion. *Algorithmica*, 84(11):3246–3275, 2022. doi:10.1007/S00453-022-01040-9.

[80] Gabriel Andrew Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25(1-2):71–76, 1961. doi:10.1007/BF02992776.

[81] Pavel Dvořák, Tomáš Masařík, Jana Novotná, Monika Krawczyk, Paweł Rzążewski, and Aneta Żuk. List locally surjective homomorphisms in hereditary graph classes. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea*, volume 248 of *LIPIcs*, pages 30:1–30:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ISAAC.2022.30.

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

[83] Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972. doi:10.1145/321694.321699.

[84] Thomas Emden-Weinert, Stefan Hougardy, and Bernd Kreuter. Uniquely colourable graphs and the hardness of colouring graphs of large girth. *Combinatorics Probability and Computing*, 7(4):375–386, 1998. doi:10.1017/S0963548398003678.

[85] Pál Erdős and Lajos Pósa. On independent circuits contained in a graph. *Canadian Journal of Mathematics*, 17:347–352, 1965. doi:10.4153/CJM-1965-035-8.

[86] Wolfgang Espelage, Frank Gurski, and Egon Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Proc. WG 2001*, pages 117–128. LNCS, 2001. doi:10.1007/3-540-45477-2_12.

[87] Tomás Feder and Carlos S. Subi. Packing edge-disjoint triangles in given graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:13, 2012. URL: http://eccc.hpi-web.de/report/2012/013.

[88] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width is np-complete. *SIAM Journal on Discrete Mathematics*, 23(2):909–939, 2009. doi:10.1137/070687256.

[89] Angèle M. Foley, Dallas J. Fraser, Chính T. Hoàng, Kevin Holmes, and Tom P. LaMantia. The intersection of two vertex coloring problems. *Graphs and Combinatorics*, 36(1):125–138, 2020. doi:10.1007/s00373-019-02123-1.

[90] Fedor V. Fomin and Tuukka Korhonen. Fast FPT-approximation of branchwidth. *SIAM Journal on Computing*, 53(4):1085–1131, 2024. doi:10.1137/22M153937X.

[91] Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and cmso. *SIAM Journal on Computing*, 44(1):54–87, 2015. doi:10.1137/140964801.

[92] Dallas J. Fraser, Angèle M. Hamel, Chính T. Hoàng, Kevin Holmes, and Tom P. LaMantia. Characterizations of $(4K_1, C_4, C_5)$-free graphs. *Discrete Applied Mathematics*, 10(1016):166–174, 2017. doi:10.1016/j.dam.2016.08.016.

[93] Jakub Gajarský, Lars Jaffke, Paloma T. Lima, Jana Novotná, Marcin Pilipczuk, Paweł Rzążewski, and Uéverton S. Souza. Taming graphs with no large creatures and skinny ladders. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 58:1–58:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.58.

[94] Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1-2):25–66, 1967.

[95] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976. doi:10.1016/0304-3975(76)90059-1.

[96] Peter Gartland and Daniel Lokshtanov. Dominated minimal separators are tame (nearly all others are feral). *CoRR*, abs/2007.08761, 2020. URL: https://arxiv.org/abs/2007.08761, arXiv:2007.08761.

[97] Peter Gartland and Daniel Lokshtanov. 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*, pages 613–624. IEEE, 2020. doi:10.1109/FOCS46700.2020.00063.

[98] Peter Gartland and Daniel Lokshtanov. Graph classes with few minimal separators. I. finite forbidden induced subgraphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3063–3097. SIAM, 2023. doi:10.1137/1.9781611977554.CH119.

[99] Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 683–691. ACM, 2024. doi:10.1145/3618260.3649791.

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

[101] Serge Gaspers, Shenwei Huang, and Daniël Paulusma. Colouring square-free graphs without long induced paths. *Journal of Computer and System Sciences*, 10(1016):60–79, 2019. doi:10.1016/j.jcss.2019.06.002.

[102] Michael U. Gerber and Daniel Kobler. Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoretical Computer Science*, 299(1):719–734, 2003. doi:10.1016/S0304-3975(02)00725-9.

[103] Jan Goedgebeur and Oliver Schaudt. Exhaustive generation of k-critical H-free graphs. *Journal of Graph Theory*, 87(2):188–207, April 2017. `doi:10.1002/jgt.22151`.

[104] Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, March 2016. `doi:10.1002/jgt.22028`.

[105] Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on H-free graphs. *Information and Computation*, 237:204–214, October 2014. `doi:10.1016/j.ic.2014.02.004`.

[106] Petr A. Golovach, Daniël Paulusma, and Jian Song. Coloring graphs without short cycles and long induced paths. *Discrete Applied Mathematics*, 167:107–120, April 2014. `doi:10.1016/j.dam.2013.12.008`.

[107] Martin Grötschel, László Lovász, and Alexander Schrijver. Polynomial algorithms for perfect graphs. In *Topics on Perfect Graphs*, pages 325–356. Elsevier, 1984. `doi:10.1016/s0304-0208(08)72943-8`.

[108] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988. `doi:10.1007/978-3-642-97881-4`.

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

[110] Frank Gurski. The behavior of clique-width under graph operations and graph transformations. *Theory of Computing Systems*, 60(2):346–376, 2017. `doi:10.1007/s00224-016-9685-1`.

[111] András Gyárfás. 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, pages 801–816. North-Holland, Amsterdam, 1975.

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

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

[114] Penny E. Haxell. Packing and covering triangles in graphs. *Discrete Mathematics*, 195(1–3):251–254, 1999. `doi:10.1016/S0012-365X(98)00183-6`.

[115] Penny E. Haxell and Vojtěch Rödl. Integer and fractional packings in dense graphs. *Combinatorica*, 21(1):13–38, January 2001. `doi:10.1007/s004930170003`.

[116] Pinar Heggernes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. *Theor. Comput. Sci.*, 511:172–180, 2013. `doi:10.1016/j.tcs.2012.03.013`.

[117] Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Jesper Nederlof. Computing the cutwidth of bipartite permutation graphs in linear time. *SIAM Journal on Discrete Mathematics*, 26(3):1008–1021, 2012. `doi:10.1137/110830514`.

[118] Pinar Heggernes, Daniel Meister, and Charis Papadopoulos. A new representation of proper interval graphs with an application to clique-width. *Electronic Notes in Discrete Mathematics*, 32:27–34, 2009. `doi:10.1016/j.endm.2009.02.005`.

[119] Pavol Hell and Shenwei Huang. Complexity of coloring graphs without paths and cycles. *Discrete Applied Mathematics*, 216:211–232, January 2017. `doi:10.1016/j.dam.2015.10.024`.

[120] Petr Hliněný, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *The Computer Journal*, 51(3):326–362, 2008. `doi:10.1093/comjnl/bxm052`.

[121] Chính T. Hoàng, Marcin Kamiński, Vadim Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, May 2008. `doi:10.1007/s00453-008-9197-8`.

[122] Chính T. Hoàng and D. Adam Lazzarato. Polynomial-time algorithms for minimum weighted colorings of $(P_5, \overline{P_5})$-free graphs and similar graph classes. *Discrete Applied Mathematics*, 10(1016):106–111, 2015. `doi:10.1016/j.dam.2015.01.022`.

[123] Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. `doi:10.1007/s00453-012-9661-3`.

[124] Ian Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720, November 1981. `doi:10.1137/0210055`.

[125] Shenwei Huang. Improved complexity results on k-coloring P_t-free graphs. *European Journal of Combinatorics*, 51:336–346, January 2016. `doi:10.1016/j.ejc.2015.06.005`.

[126] Shenwei Huang, Matthew Johnson, and Daniël Paulusma. Narrowing the complexity gap for colouring (C_s, P_t)-free graphs. *The Computer Journal*, 58(11):3074–3088, June 2015. `doi:10.1093/comjnl/bxv039`.

[127] Shenwei Huang and T. Karthick. On graphs with no induced five-vertex path or paraglider. *Journal of Graph Theory*, 97(2):305–323, 2021. `doi:10.1002/jgt.22656`.

[128] Vít Jelínek, Tereza Klimošová, Tomáš Masařík, Jana Novotná, and Aneta Pokorná. On 3-coloring of $(2P_4, C_5)$-free graphs. In *Graph-Theoretic Concepts in Computer Science (WG)*, pages 388–401. Springer International Publishing, 2021. doi:10.1007/978-3-030-86838-3_30.

[129] Vít Jelínek, Tereza Klimošová, Tomáš Masařík, Jana Novotná, and Aneta Pokorná. On 3-coloring of $(2P_4, C_5)$-free graphs. *Algorithmica*, 84(6):1526–1547, 2022. doi:10.1007/S00453-022-00937-9.

[130] Marcin Kamiński, Vadim V. Lozin, and Martin Milanič. Recent developments on graphs of bounded clique-width. *Discrete Applied Mathematics*, 157(12):2747–2761, 2009. doi:10.1016/j.dam.2008.08.022.

[131] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: 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*, pages 85–103. Springer US, Boston, MA, 1972. doi:10.1007/978-1-4684-2001-2_9.

[132] David Kelly. The 3-irreducible partially ordered sets. *Can. J. of Math.*, 29:367–383, 1977.

[133] Sándor Kisfaludi-Bak, Jana Masaříková, Erik Jan van Leeuwen, Bartosz Walczak, and Karol Węgrzycki. Separator theorem and algorithms for planar hyperbolic graphs. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry, SoCG 2024, June 11-14, 2024, Athens, Greece*, volume 293 of *LIPIcs*, pages 67:1–67:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.SOCG.2024.67.

[134] Tereza Klimošová, Josef Malík, Tomáš Masařík, Jana Novotná, Daniël Paulusma, and Veronika Slívová. Colouring $(P_r + P_s)$-Free Graphs. *Algorithmica*, 82(7):1833–1858, January 2020. doi:10.1007/s00453-020-00675-w.

[135] Daniel Kobler and Udi Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003. doi:10.1016/S0166-218X(02)00198-1.

[136] Jan Kratochvíl, Tomáš Masařík, and Jana Novotná. U-bubble model for mixed unit interval graphs and its applications: The MaxCut problem revisited. *Algorithmica*, 83(12):3649–3680, 12 2021. doi:10.1007/s00453-021-00837-4.

[137] Stefan Kratsch and Pascal Schweitzer. Graph isomorphism for graph classes characterized by two forbidden induced subgraphs. *Discrete Applied Mathematics*, 10(1016):240–253, 2017. doi:10.1016/j.dam.2014.10.026.

[138] Michael Krivelevich. On a conjecture of tuza about packing and covering of triangles. *Discrete Mathematics*, 142(1):281–286, 1995. doi:10.1016/0012-365X(93)00228-W.

[139] Melven R. Krom. The decision problem for a class of first-order formulas in which all disjunctions are binary. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 13(1-2):15–20, 1967. doi:10.1002/malq.19670130104.

[140] Kai-Yuan Lai, Hsueh-I Lu, and Mikkel Thorup. Three-in-a-tree in near linear time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1279–1292. ACM, 2020. doi:10.1145/3357713.3384235.

[141] S. Aparna Lakshmanan, Csila Bujtás, and Zsolt Tuza. Small edge sets meeting all triangles of a graph. *Graphs and Combinatorics*, 28(3):381–392, April 2011. doi:10.1007/s00373-011-1048-8.

[142] C.G. Lekkerkerker and J.Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1):45–64, 1962.

[143] Daniel Leven and Zvi Galil. NP completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35–44, March 1983. doi:10.1016/0196-6774(83)90032-9.

[144] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

[145] Daniel Lokshantov, Martin Vatshelle, and Yngve Villanger. Independent set in P_5-free graphs in polynomial time. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 570–581. SIAM, 2014. doi:10.1137/1.9781611973402.43.

[146] Daniel Lokshtanov. Wheel-free deletion is $W[2]$-hard. In Martin Grohe and Rolf Niedermeier, editors, *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*, volume 5018 of *Lecture Notes in Computer Science*, pages 141–147. Springer, 2008. doi:10.1007/978-3-540-79723-4_14.

[147] Vadim V. Lozin and Dieter Rautenbach. On the band-, tree-, and clique-width of graphs with bounded vertex degree. *SIAM Journal on Discrete Mathematics*, 18(1):195–206, 2004. doi:10.1137/S0895480102419755.

[148] Vadim V. Lozin and Dieter Rautenbach. The tree- and clique-width of bipartite graphs in special classes. *Australasian Journal of Combinatorics*, 34:57–67, 2006. URL: https://ajc.maths.uq.edu.au/pdf/34/ajc_v34_p057.pdf.

[149] Vadim V. Lozin and Jordan Volz. The clique-width of bipartite graphs in monogenic classes. *International Journal of Foundations of Computer Science*, 19(02):477–494, 2008. doi:10.1142/S0129054108005772.

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

[151] Konrad Majewski, Tomáš Masařík, Jana Novotná, Karolina Okrasa, Marcin Pilipczuk, Paweł Rzążewski, and Marek Sokołowski. Max weight independent set in graphs with no long claws: An analog of the Gyárfás' path argument. In Mikołaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4–8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 93:1–93:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.93.

[152] Johann A. Makowsky and Udi Rotics. On the clique-width of graphs with few P_4's. *International Journal of Foundations of Computer Science*, 10(03):329–348, 1999. doi:10.1142/S0129054199000241.

[153] Dmitriy S. Malyshev and O. O. Lobanova. Two complexity results for the vertex coloring problem. *Discrete Applied Mathematics*, 10(1016):158–166, 2017. doi:10.1016/j.dam.2016.10.025.

[154] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. doi:10.1007/s00453-008-9233-8.

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

[156] Stephan Olariu. Paw-free graphs. *Information Processing Letters*, 28(1):53–54, 1988. doi:10.1016/0020-0190(88)90143-3.

[157] Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

[158] Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Quasi-polynomial-time algorithm for independent set in P_t-free graphs via shrinking the space of induced paths. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 204–209. Society for Industrial and Applied Mathematics, January 2021. doi:10.1137/1.9781611976496.23.

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

[160] Amir Pnueli, Abraham Lempel, and Shimon Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23(1):160–175, 1971.

[161] Gregory J. Puleo. Tuza's conjecture for graphs with maximum average degree less than 7. *European Journal of Combinatorics*, 49:134–152, 2015. doi:10.1016/j.ejc.2015.03.006.

[162] Michaël Rao. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theoretical Computer Science*, 377(1–3):260–267, 2007. doi:10.1016/j.tcs.2007.03.043.

[163] Jean-Florent Raymond. Dynamic Erdős-Pósa listing. Available at https://perso.limos.fr/~jfraymon/Erd%C5%91s-P%C3%B3sa/.

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

[165] Pascal Schweitzer. Towards an isomorphism dichotomy for hereditary graph classes. *Theory of Computing Systems*, 61(4):1084–1127, 2017. doi:10.1007/s00224-017-9775-8.

[166] Jeremy P. Spinrad, Andreas Brandstädt, and Lorna Stewart. Bipartite permutation graphs. *Discrete Applied Mathematics*, 18(3):279–292, 1987. doi:10.1016/S0166-218X(87)80003-3.

[167] Sophie Spirkl, Maria Chudnovsky, and Mingxian Zhong. Four-coloring P_6-free graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1239–1256. Society for Industrial and Applied Mathematics, January 2019. doi:10.1137/1.9781611975482.76.

[168] Robert Endre Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221–232, 1985. doi:10.1016/0012-365X(85)90051-2.

[169] William T. Trotter and John I. Moore Jr. Characterization problems for graphs, partially ordered sets, lattices, and families of sets. *Discrete Mathematics*, 16(4):361–381, 1976. doi:10.1016/S0012-365X(76)80011-8.

[170] Zsolt Tuza. A conjecture: Finite and infinite sets, Eger, Hungary 1981, A. Hajnal, László Lovász, Vera T. Sós. In *Proc. Colloq. Math. Soc. J. Bolyai*, volume 37, page 888, 1981.

[171] Martin Vatshelle. *New Width Parameters of Graphs.* PhD thesis, University of Bergen, 2012.

[172] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM Journal on Computing*, 38(5):2007–2020, 2009. doi:10.1137/070710913.

[173] David Zuckerman. Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007. doi:10.4086/toc.2007.v003a006.