


Numéro National de Thèse : **2018LYSEN021**

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée par
l'Ecole Normale Supérieure de Lyon

Ecole Doctorale N° 512
École Doctorale en Informatique et Mathématiques de Lyon

Spécialité de doctorat : Théorie des graphes
Discipline : Informatique

Soutenue publiquement le 08/06/2018, par :
Ngoc Khang LE

Detecting and coloring some graph classes

Détection et coloration de certaines classes de graphes

Devant le jury composé de :

Mme. Maria Chudnovsky	Professeure, Princeton University	Rapporteure
M. Paul Dorbec	Maître de conférences, Université de Bordeaux	Rapporteur
M. Gwenael Joret	Professeur assistant, Université libre de Bruxelles	Examinateur
Mme. Myriam Preissmann	Directrice de recherche, Université Grenoble Alpes	Examinatrice
M. Nicolas Trotignon	Directeur de recherche, ENS de Lyon	Directeur de thèse

Contents

Introduction (in French) **3**

Introduction **7**

1 Basic notions **11**
1.1 Graphs . . . 11
1.2 Perfect graphs and χ-boundedness . . . 14
1.3 Some kinds of decomposition and graph operations . . . 15
1.4 Some particular graphs . . . 17

2 IKS4-free graphs **19**
2.1 Chromatic number of ISK4-free graphs . . . 19
2.1.1 Introduction . . . 19
2.1.2 Preliminaries . . . 21
2.1.3 Proof of Theorem 2.1.7 . . . 24
2.1.4 Proof of Theorem 2.1.8 . . . 27
2.1.5 Conclusion . . . 32
2.2 Detecting an ISK4 . . . 32
2.2.1 Introduction . . . 32
2.2.2 Preliminaries . . . 33
2.2.3 Antennas and cables . . . 34
2.2.4 Proof of Lemma 2.2.2 . . . 45
2.2.5 Conclusion . . . 46

3 Even-hole-free graphs **47**
3.1 Introduction . . . 47
3.2 Preliminaries . . . 49
3.3 χ-bounding function . . . 53
3.3.1 Special graphs . . . 53

3.3.2 Elimination order 54
3.3.3 The bound is tight 59
3.4 Rank-width 60
3.4.1 Bounded rank-width 60
3.4.2 An even-hole-free graph with no clique cutset and unbounded rank-width 67

4 Connected greedy coloring **69**
4.1 Introduction 70
4.2 Some properties of minimally bad graphs 74
4.3 Forbidden structures of minimally bad graphs 79
4.4 Proof of Theorem 4.1.3 88
4.5 Conclusion 95

Conclusion **97**

Contribution **99**

Bibliography **105**

Introduction (in French)

Les graphes sont des structures mathématiques utilisées pour modéliser les relations par paires entre objets. Plus formellement, un *graphe* est une paire ordonnée $G = (V, E)$, où V est un ensemble de *sommets* (ou *noeuds*) et E est un ensemble d'*arêtes*, qui sont des sous-ensembles à 2 éléments de V. Un graphe peut donc être dessiné en spécifiant un ensemble de points et en reliant ces paires qui forment des arêtes. Un exemple de graphe est donné ci-dessous.

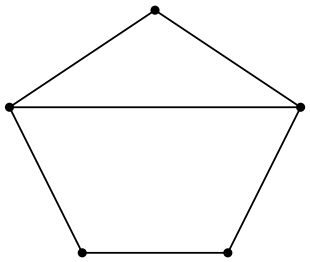


Figure 1: Un exemple de graphe

Malgré leur structure simple, les graphes ont des applications dans divers domaines tels que l'informatique, la physique, la biologie et la sociologie. Par exemple, un graphe peut être utilisé pour modéliser une carte de transport public, où chaque arrêt est un sommet et deux arrêts conscutifs de la même ligne forment une arête. Un autre exemple est le réseau social, où chaque sommet représente une personne et où il existe une arête entre deux personnes si elles se connaissent. La théorie des graphes implique de répondre à plusieurs problèmes concernant certains graphes spécifiques pour comprendre leur structure, ce qui peut conduire à de nombreuses applications dans la vie réelle. Deux problèmes parmi eux, très célèbres et importants, sont la coloration de graphe et la reconnaissance de graphe (ou la détection de graphe).

La coloration de graphe

Dans le problème de coloration, nous voulons assigner une couleur à chaque sommet telle que deux sommets reçoivent des couleurs différentes s'ils sont connectés par une arête, et le nombre de couleurs que nous utilisons est minimum. Ce problème provient d'une question sur la coloration des cartes, à savoir le problème des quatre couleurs proposée par Guthrie en 1852: supposons que vous ayez une carte divisée en régions, de combien de couleurs avez-vous besoin pour colorer chaque région de telle sorte que deux régions qui partagent une frontière commune ne reçoivent pas la même couleur? Il a conjecturé que quatre couleurs étaient toujours suffisantes. Bien que cette question ait suscité beaucoup d'attention de la part des mathématiciens, elle est restée non résolue jusqu'en 1976, quand Appel et Haken ont finalement donné une preuve. Il est clair que quatre couleurs ne sont jamais assez pour colorer les graphes généraux, et en fait le nombre de couleurs dont nous avons besoin peut être arbitrairement grand (par exemple, un graphe de n sommets tel qu'il y a une arête entre chaque paire d'entre eux, a besoin d'au moins n couleurs). Le problème de coloration des graphes est également difficile en le sens qu'il est NP-difficile (une classe de problèmes où aucun algorithme polynomial n'est connu, et de plus, on soupçonne qu'il n'en existe pas). Par conséquent, on peut se demander s'il existe un algorithme en temps polynomial pour colorer certaines classes de graphes.

En 1960, Berge a défini la classe des graphes parfaits et formulé une autre conjecture sur la coloration des graphes, la Conjecture Forte des Graphes Parfaits, qui concerne une caractérisation des graphes parfaits, initialement motivée par la notion de Shannon de la capacité d'erreur zéro d'un graphe. En 1984, Grötschel, Lovász et Schijver ont montré que des graphes parfaits peuvent être colorés en temps polynomial. Cependant, leur algorithme implique l'utilisation de la méthode des ellipsoïdes, qui n'est pas très efficace et peu pratique. L'existence d'un algorithme purement combinatoire pour colorer des graphes parfaits n'est pas encore connue. Malgré les efforts de nombreux scientifiques, la Conjecture Forte des Graphes Parfaits de Berge sur des graphes parfaits est restée ouverte pendant plus de 40 ans. En 2002, elle a été résolue par Chudnovsky, Robertson, Seymour et Thomas. Elle est maintenant connue comme le Théorème Fort des Graphes Parfaits. Leur preuve est basée sur un théorème de décomposition structurale profond.

La reconnaissance de graphe

Le deuxième problème que nous aimerions mentionner est la reconnaissance des graphes. Ce n'est pas un problème particulier, mais un concept très général dans la théorie des graphes. Fondamentalement, nous demandons s'il est possible de trouver

un algorithme efficace pour reconnaître certaines classes de graphes (par exemple, biparti, planaire, parfait, ...) ou pour détecter certaines structures spécifiques du graphe d'entrée. Un graphe est un *sous-graphe* de G s'il peut être obtenu de G en supprimant certains sommets et arêtes. D'un autre côté, un graphe est un sous-graphe *induit* de G s'il peut être obtenu de G en supprimant certains sommets. Par exemple, pour tout graphe fixe H, on peut vérifier en temps polynomial si un graphe G contient H comme un sous-graphe (induit). La question devient plus difficile si nous demandons si G contient une *subdivision* de H comme un sous-graphe (induit) (une *subdivision* de H est obtenue de H en subdivisant ses arêtes, la définition plus formelle est donnée dans le Chapitre 1). La différence entre le sous-graphe et le sous-graphe induit par rapport à cette question est remarquable. Dans les années 1980, le Graph Minor Project de Robertson et Seymour affirmait que pour la notion de sous-graphe, il est toujours possible de trouver un algorithme en temps polynomial. D'autre part, pour la notion de sous-graphe induit, on sait que pour certains graphes H, le problème peut être résolu en temps polynomial et pour d'autres graphes, il devient NP-difficile (voir [37] par exemple). De plus, cette dichotomie est encore loin d'être complète. L'étude des graphes parfaits a également conduit à plusieurs questions sur la détection des sous-graphes induits. Suite à la preuve du Théorème Fort des Graphes Parfaits, un algorithme en temps polynomial pour reconnaître des graphes parfaits a été découvert par Chudnovsky, Cornuéjols, Liu, Seymour et Vušković. La complexité de la détection de certains types de décomposition et de certaines structures spécifiques utilisées dans la preuve du Théorème Fort des Graphes Parfaits (par exemple, les configurations de Truemper [23]) est également considérée.

L'objectif principal de ce travail est de continuer l'étude des problèmes de coloration et de détection dans le cadre de classes de graphes fermées par sous-graphes induits (que nous appelons classes de graphes *héréditaires*).

La résumé du document

Voyons brièvement le contenu de cette thèse.

Dans le Chapitre 1, nous rappelons quelques notions de base en théorie des graphes, ainsi que certaines structures et décompositions particulières dans le contexte des sous-graphes induits que nous utilisons tout au long de la thèse. Les lecteurs qui connaissent le sujet peuvent passer ce chapitre. Dans chacun des chapitres suivants (du Chapitre 2 au Chapitre 4), nous donnons d'abord une courte introduction du problème et quelques définitions spécifiques qui ne sont utilisées que dans ce chapitre, puis nous montrons la preuve de nos résultats et éventuellement d'autres discussions.

L'objet principal de Chapitre 2 est l'étude des graphes *sans ISK4* - les graphes qui ne contiennent aucune subdivision de K_4 en tant que sous-graphe induit. Le premier résultat de ce chapitre est présenté dans la Section 2.1. Nous montrons que le nombre chromatique de cette classe est limité à 24, une amélioration considérable par rapport à la borne existant précédemment mentionnée dans [38]. Nous donnons aussi une meilleure borne dans le cas sans triangle: le nombre chromatique des graphes sans ISK4 et sans triangle est limité à 4. Notez que, juste après nos résultats, Chudnovsky *et al.* [12] améliore la limite dans le cas sans triangle à 3 en utilisant la technique de décomposition. Cependant, notre preuve est beaucoup plus simple et donne encore une belle propriété structurelle de la classe. Dans la Section 2.2, le deuxième résultat est présenté: nous montrons qu'il existe un algorithme en temps polynomial pour détecter cette classe de graphes, qui répond à une question de Chudnovsky *et al.* [15] et Lévêque *et al.* [37].

Le Chapitre 3 est consacré à l'étude de graphes sans trou pair, une classe de graphes proche des graphes parfaits. Plus spécifiquement, nous travaillons sur les graphes sans trou pair et sans étoile d'articualtion. Nous donnons une limite supérieure optimale de son nombre chromatique en termes de nombre de clique et un algorithme en temps polynomial pour les colorer. Ce dernier est, en fait, une conséquence directe de notre preuve que cette classe a rank-width bornée.

Dans le Chapitre 4, nous étudions les graphes sans griffes qui se colorient pour n'importe quel ordre connexe glouton. Ce problème est motivé par le point de vue algorithmique du problème de coloration. L'une des stratégies les plus connues pour colorier un graphe consiste à utiliser un algorithme glouton: on considère un ordre des sommets $v_1, v_2, \ldots, v_n$ du graphe d'entrée et on assigne à v_i la plus petite couleur disponible qui n'est pas utilisée par les voisins de v_i parmi $v_1, v_2, \ldots, v_{i-1}$ (en ajoutant une nouvelle couleur si nécessaire). Le nombre de couleurs que nous utilisons par cet algorithme dépend fortement de l'ordre choisi. Ici nous considérons seulement les ordres *connexes*, c'est-à-dire où chaque v_i a au moins un voisin parmi $v_1, v_2, \ldots, v_{i-1}$. Un graphe G est *bon* si pour chaque sous-graphe induit connexe H de G, chaque ordre connexe donne H une coloration optimale. Nous donnons la caractérisation complète de bons graphes sans griffes en termes de sous-graphes induits minimaux interdits.

Introduction

Graphs are mathematical structures used to model pairwise relations between objects. More formally, a *graph* is an ordered pair $G = (V, E)$, where V is a set of *vertices* (or *nodes*) and E is a set of *edges*, which are 2-element subsets of V. A graph therefore can be drawn just by specifying a set of points and connecting those pairs which form edges. An example of a graph is given below.

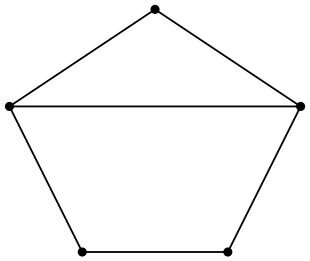


Figure 2: An example of a graph

Despite their simple structure, graphs have applications in various areas like computer science, physics, biology and sociology. For example, a graph can be used to model a public transport map, where each stop is a vertex and two consecutive stops of the same line form an edge. Another example is the social network, where each vertex represents a person and there exists an edge between two people if they know each other. Graph theory involves answering several problems concerning some specific graphs to understand their structures, which might lead to many real-life applications. Two problems among them which are very famous and important are graph coloring and graph recognition (or graph detection).

Graph coloring

In coloring problem, we would like to assign for each vertex a color such that two vertices receive different colors if they are connected by an edge, and the number

of colors that we use is minimum. This problem originates from a question about coloring of maps, namely the Four-Color Conjecture proposed by Guthrie in 1852: suppose that you have a map divided into regions, how many colors do you need to color each region such that no two regions that share a common border receive the same color? He conjectured that four colors were always enough. Although this question attracted a lot of attention from mathematicians, it remained unsolved until 1976, when Appel and Haken finally gave a proof. It is clear that four colors are never enough to color general graphs, and in fact the actual number of colors that we need might be arbitrarily large (for instance, a graph of n vertices such that there is an edge between every pair of them, needs at least n colors). The graph coloring problem is also proved to be difficult in a sense that it is NP-hard (a class of problems where no polynomial-time algorithm is known, and moreover, it is suspected that there does not exist one). Therefore, one might wonder whether there exists a polynomial-time algorithm to color certain graph classes.

In 1960, Berge defined the class of perfect graphs and formulated another conjecture about graph coloring, the Strong Perfect Graph Conjecture, which concerns a characterization of perfect graphs, originally motivated by Shannon's notion of the zero-error capacity of a graph. In 1984, Grötschel, Lovász and Schijver showed that perfect graphs can be colored in polynomial time. However, their algorithm involves the use of ellipsoid method, which is not very efficient and impractical. The existence of a purely combinatorial algorithm to color perfect graphs is still not known. Despite the efforts of many scientists, the Berge's Strong Perfect Graph Conjecture on perfect graphs remained open for more than 40 years. Until 2002, it was settled by Chudnovsky, Robertson, Seymour and Thomas. It is now known as the Strong Perfect Graph Theorem. Their proof based on a deep structural decomposition theorem.

Graph recognition

Second problem that we would like to mention is graph recognition. This is not a particular problem, but a very general concept in graph theory. Basically, we ask whether it is possible to find an efficient algorithm to recognize certain graph classes (e.g. bipartite, planar, perfect,...) or to detect some specific structures of the input graph. A graph is a *subgraph* of G if it can be obtained from G by deleting some vertices and edges. On the other hand, a graph is an *induced subgraph* of G if it can be obtained from G by deleting some vertices. For example, for any fixed graph H, one can check in polynomial time whether a graph G contains H as a (induced) subgraph. The question becomes more difficult if we ask if G contains

a *subdivision* of H as a (induced) subgraph (a *subdivision* of H is obtained from H by subdividing its edges, more formal definition is given in Chapter 1). The difference between subgraph and induced subgraph with respect to this question are remarkable. In the 1980s, the Graph Minor Project by Robertson and Seymour asserted that for subgraph, it is always possible to find a polynomial algorithm. On the other hands, for induced subgraph, we know that for some graphs H, the problem is polynomial-time solvable and for some other graphs, it becomes NP-hard (see [37] for example). Moreover, this dichotomy is still very far from being complete. The study of perfect graphs also led to several questions on detecting induced subgraphs. Subsequent to the proof of Strong Perfect Graph Theorem, a polynomial algorithm for recognizing perfect graphs was discovered by Chudnovsky, Cornuéjols, Liu, Seymour and Vušković. The complexity of detecting certain kinds of decomposition and some specific structures used in the proof of Strong Perfect Graph Theorem (for example, Truemper configurations [23]) is also considered.

The main focus of this thesis is to continue the study of the coloring and detecting problem in the setting of graph classes closed under taking induced subgraphs (which we call *hereditary* graph classes).

Outline of the document

Let us briefly review the content of this thesis.

In Chapter 1, we recall some basic notions in graph theory, as well as some particular structures and decompositions in the context of induced subgraph that we use throughout the thesis. The readers who are familiar with the subject can skip this chapter. In each of the following chapters (from Chapter 2 to Chapter 4), we first give a short introduction of the problem and some specific definitions that are only used in that chapter, and then we show the proof of our results and possibly some further discussions.

The main object of Chapter 2 is *ISK4-free* graphs - the graphs that do not contain any subdivision of K_4 as an induced subgraph. The first result of this chapter is presented in the Section 2.1. We prove that the chromatic number of this class is bounded by 24, a big improvement compared to the best known bound which was mentioned in [38]. We give also a much better bound in triangle-free case: the chromatic number of (triangle, ISK4)-free graphs is bounded by 4. Note that, right after our results, Chudnovsky *et al.* [12] improve the bound in triangle-free case to 3 by using decomposition technique. However, our proof is much simpler and still gives a nice structural property of the class. In Section 2.2, the second result is presented: we prove that there exists a polynomial-time algorithm for detecting this

graph class, which answers a question by Chudnovsky *et al.* [15] and Lévêque *et al.* [37].

Chapter 3 is devoted to the study of even-hole-free graphs, a graph class which is close to perfect graphs. More specifically, we work on even-hole-free graphs with no star cutset. We give an optimal upper bound for its chromatic number in terms of clique number and a polynomial-time algorithm to color them. The latter is, in fact, a direct consequence of our proof that this class has bounded rank-width.

In Chapter 4, we study claw-free graphs which are good to color with respect to any greedy connected order. This problem is motivated by algorithmic point of view of coloring problem. One of the most well-known strategy to color a graph is by using a greedy algorithm: we consider an order of the vertices $v_1, v_2, \ldots, v_n$ of the input graph and assign for v_i the smallest available color which is not used by v_i's neighbors among $v_1, v_2, \ldots, v_{i-1}$ (adding a new color if needed). The number of color we use by this algorithm heavily depends on the chosen order. Here we consider only *connected* order, an order where each v_i has at least a neighbor among $v_1, v_2, \ldots, v_{i-1}$. A graph G is *good* if for every connected induced subgraph H of G, every connected order gives H an optimal coloring. We give the complete characterization of good claw-free graphs in terms of minimal forbidden induced subgraphs.

Chapter 1

Basic notions

In the first section of this chapter, we recall some basic notions of graph theory that we use throughout the thesis. Most of them are standard and the reader is referred to [22] for any undefined terms. In the following sections, we introduce several more specific definitions: the notion of χ-boundedness, some kinds of decomposition and some particular graphs. The readers who are familiar with this field can also skip these sections.

1.1 Graphs

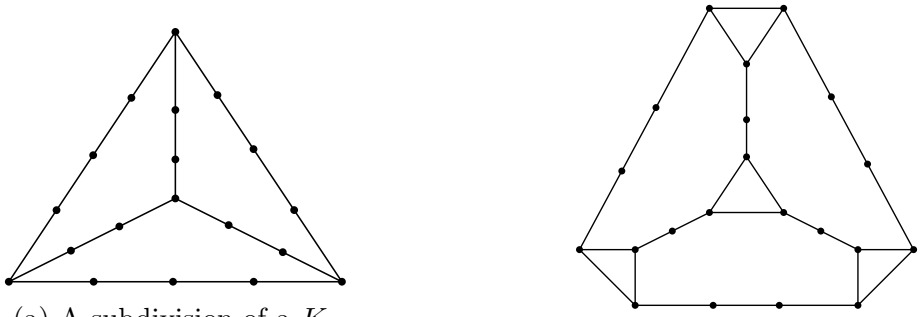
A *graph* G is an ordered pair (V, E) consisting of a vertex (node) set V and an edge set $E \subseteq \binom{V}{2}$, where $\binom{V}{2}$ is the set of all the subsets of size 2 of V. An element $\{u, v\}$ of E is also denoted by uv or vu. We also refer to V and E as $V(G)$ and $E(G)$, respectively. Two vertices u and v in $V(G)$ such that $uv \in E(G)$ are said to be *adjacent*, and we also say that u is a neighbor of v. The (*open*) *neighborhood* of a vertex v, denoted by $N_G(v)$, is the set of neighbors of v. The *closed neighborhood* of v, denoted by $N_G[v]$, is defined as $N_G(v) \cup \{v\}$. The *degree* of v in G is $|N_G(v)|$. We also extend this notion for a subset $X \subseteq V(G)$ by defining $N_G(X) := (\cup_{v \in X} N_G(v)) \setminus X$ and $N_G[X] := N_G(X) \cup X$. We might also write $N(v)$, $N[v]$, $N(X)$, $N[X]$ instead of $N_G(v)$, $N_G[v]$, $N_G(X)$, $N_G[X]$, respectively, if there is no ambiguity. For some subsets $K \subseteq V(G)$ and $C \subseteq V(G) \setminus K$, we denote by $N_K(C)$ the set of neighbors of C in K, or $N_K(C) := N(C) \cap K$. The *complement* of G is the graph $\overline{G} = (V, \overline{E})$, where $\overline{E} = \binom{V}{2} \setminus E$. A *clique* in G is a set of vertices that are pairwise adjacent. A *stable set* (or an *independent set*) in G is a set of vertices that are pairwise non-adjacent. The *size* of a clique or a stable set is its number of vertices. Let A and B be two disjoint subsets of $V(G)$, we say that A is *complete* to B if for every $u \in A$ and every $v \in B$,

$uv \in E(G)$. If the set A is of size 1, say $A = \{u\}$, we also say that u is complete to B (instead of saying that $\{u\}$ is complete to B). The *complete graph* K_n is the graph on n vertices that are pairwise adjacent. A graph is called *complete bipartite* (resp. *complete tripartite*) if its vertex set can be partitioned into two (resp. three) non-empty stable sets that are pairwise complete to each other. If these two (resp. three) sets have size p, q (resp. p, q, r) then the graph is denoted by $K_{p,q}$ (resp. $K_{p,q,r}$, see Figure 1.1 for example). Given a graph H, the *line graph* of H, denoted by $L(H)$, is the graph with vertex set $E(H)$ and edge set $\{ef : e \cap f \neq \emptyset\}$. The *edge subdivision operation* for an edge $uv \in E(G)$ is the deletion of uv from G and the addition of two edges uw and wv along with the new vertex w. This operation generates a new graph $H = (V \cup \{w\}, (E \setminus \{uv\}) \cup \{uw, wv\})$. A *subdivision* of G is a graph that can be derived from G by a sequence of edge subdivision operations. See Figure 1.2 for an example of subdivisions and line graphs.



(a) $K_{3,3}$ (b) $K_{2,2,2}$

Figure 1.1: An example of a complete bipartite graph and a complete tripartite graph

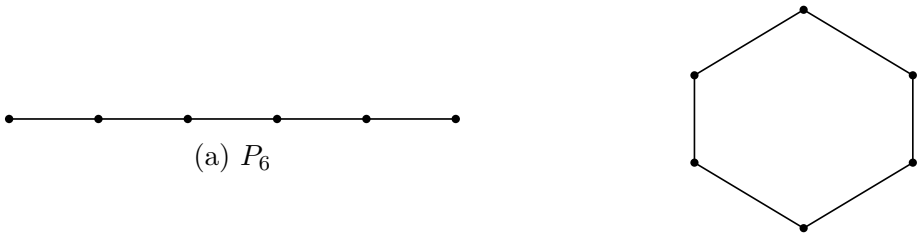


(a) A subdivision of a K_4 (b) Its line graph

Figure 1.2: An example of subdivisions and line graphs

A *path* P on k vertices ($k \geq 1$) is a graph with vertex set $\{p_1, \ldots, p_k\}$ such that either $k = 1$, or for $i, j \in \{1, \ldots, k\}$, p_i is adjacent to p_j if and only if $|i - j| = 1$

(this is often referred to as *induced path* or *chordless path* in literature). We call p_1 and p_k the *ends* of the path P. The *interior* of P is $\{p_2, \ldots, p_{k-1}\}$ and each vertex in the interior of P is said to be an *interior* (*internal*) vertex of P. Let $u, v \in V(P)$, we denote by uPv the subpath of P from u to v and denote by P^* the subpath of P from p_2 to p_{k-1} (i.e. $P^* := p_2 P p_{k-1}$). A path P is *flat* in G if all the interior vertices of P are of degree 2 in G. The *length* of a path on k vertices is $(k-1)$. A path on k vertices is denoted by P_k. A *cycle* C on k vertices ($k \geq 3$) is a graph with vertex set $\{p_1, \ldots, p_k\}$ such that for $i, j \in \{1, \ldots, k\}$, p_i is adjacent to p_j if $|i-j| = 1$, where the index is taken modulo k. All the other edges of C (the edges $p_i p_j$ such that $|i-j| > 1$) are its *chords*. A cycle is said to be *chordless* (or *induced*) if it has no chord. The *length* of a cycle on k vertices is k. A *hole* is a chordless cycle of length at least 4. A hole of length k is called a *k-hole* and is denoted by C_k. A hole is said to be *odd* (resp. *even*) if its length is odd (resp. even). An *antihole* is the complement of a hole. Note that K_3 is a chordless cycle but it is not a hole. The *girth* of a graph is the smallest length of its cycle. See Figure 1.3 for an example of paths and holes.



(a) P_6

(b) C_6

Figure 1.3: An example of paths and holes

Let $G = (V, E)$, $G' = (V', E')$. We say that G is *isomorphic* to G' if there exists a bijection $\phi : V \to V'$ such that $uv \in E \iff \phi(u)\phi(v) \in E'$. Graph G' is said to be a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. For some graph H, we say that graph G *contains H as a subgraph* if there exists $X \subseteq V$ and $Y \subseteq E$ such that (X, Y) is isomorphic to H. If $X \subseteq V$, the *subgraph of G induced by X*, denoted by $G[X]$, is (X, E_X) where $E_X = \{uv | uv \in E \text{ and } u, v \in X\}$. A subgraph G' of G is said to be an *induced subgraph* of G if there exists $X \subseteq V$ such that $G' = G[X]$. For convenience, the subgraph of G induced by $V \setminus X$ is often denoted by $G \setminus X$ instead of $G[V \setminus X]$. For $v \in V$, we also write $G \setminus v$ instead of $G \setminus \{v\}$. For some graph H, we say that graph G *contains H as an induced subgraph* if there exists $X \subseteq V$ such that $G[X]$ is isomorphic to H. A graph is *H-free* if it does not contain H as an induced subgraph. Given a family of graphs $\mathcal{H}$, a graph G is said to be $\mathcal{H}$-free

if it is H-free for every $H \in \mathcal{H}$. A class of graphs is *hereditary* if it is closed under taking induced subgraphs. It is clear that a graph class defined by forbidding a list of graphs is hereditary. Since hereditary graph classes are the main focus of our work, for convenience, we often say that a graph G *contains* a graph H if G contains H as an induced subgraph.

1.2 Perfect graphs and χ-boundedness

Let us first introduce three graph parameters which play an important role in graph theory. Computing any of them for general graphs is well-known to be NP-hard [33].

- The *chromatic number* of G, denoted by $\chi(G)$, is the smallest number of colors needed to color the vertices of G so that no two adjacent vertices receive the same color (such a coloring is called *proper*). Equivalently, one can define the chromatic number of G as the smallest number k such that $V(G)$ can be partitioned into k stable sets. A coloring of G with $\chi(G)$ colors is called an *optimal coloring* of G.
- The *clique number* of G, denoted by $\omega(G)$, is the size of a largest clique in G.
- The *stability number* (or *independence number*) of G, denoted by $\alpha(G)$, is the size of a largest stable set in G.

It is clear that $\chi(G) \geq \omega(G)$, since a clique of size k needs at least k different colors in any proper coloring of G. Therefore, clique number gives a natural lower bound for chromatic number. The question is: when does the equality hold? It is clear that the equality does not hold for every graph since a hole of length 5 has $\omega(C_5) = 2$, but $\chi(C_5) = 3$. In fact, the gap between $\chi(G)$ and $\omega(G)$ can be arbitrarily large. A *triangle* is a graph isomorphic to a K_3. In 1955, Mycielski gave a construction of a family of triangle-free graphs which has large chromatic number [39]. Later, using probabilistic method, Erdős even generalized this result by proving the existence of a graph whose girth and chromatic number are both large [25].

Perfect graphs are defined in such a way that this equality holds for them. A graph G is *perfect* if for every induced subgraph H of G, $\chi(H) = \omega(H)$. In Berge's initial work on perfect graph, he made some nice conjectures on their structure [5]. The most important one, called Strong Perfect Graph Conjecture, was proved by Chudnovsky, Robertson, Seymour and Thomas around 40 years later [13]. An *odd antihole* is the complement of an odd hole. The graphs that do not contain an odd hole nor an odd antihole as an induced subgraph are known as *Berge* graphs.

Theorem 1.2.1 (Strong Perfect Graph Theorem [13]). *A graph is perfect if and only if it is a Berge graph.*

The concept of χ-boundedness was introduced by Gyárfás [30] as a natural extension of perfect graphs: a class of graph $\mathcal{G}$ is *χ-bounded* with *χ-bounding function* f if, for every graph $G \in \mathcal{G}$, $\chi(G) \leq f(\omega(G))$. It is clear that the class of perfect graphs is χ-bounded with the χ-bounding function $f(x) = x$. Now the question is: which induced subgraphs need to be forbidden to get a χ-bounded class of graphs? The previously mentioned result of Erdős [25] implies that forbidding only one induced subgraph H may lead to a χ-bounded class only if H has no cycle. Gyárfás conjectured that this is sufficient:

Conjecture 1.2.2 (Gyárfás [30]). *The class of H-free graphs is χ-bounded if H is a forest.*

Another way to obtain a χ-bounded class is to forbid infinite number of graphs. A lot of questions on the χ-boundedness of graph classes defined by forbidding odd holes, even holes, long holes,... have been asked [30]. In this thesis, we consider this problem on ISK4-free graphs (see Chapter 2) and even-hole-free graphs with no star cutset (see Chapter 3).

1.3 Some kinds of decomposition and graph operations

In this section, we define some kind of decomposition that we use throughout the thesis. Let $G = (V, E)$ be a graph. A *cutset* in graph G is a subset $S \subsetneq V(G)$ such that $G \setminus S$ is disconnected.

- For any $k \geq 0$, a *k-cutset* is a cutset of size k.
- The only vertex of a 1-cutset is called the *cut-vertex*.
- A cutset S is a *star cutset* if S contains a node x adjacent to every node in $S \setminus x$.
- A cutset S is a *clique cutset* if S is a clique. It is clear that clique cutset is a particular star cutset.
- A *proper* 2*-cutset* is a 2-cutset $\{a, b\}$ such that $ab \notin E$, $V(G) \setminus \{a, b\}$ can be partitioned into two non-empty sets X and Y so that there is no edge between X and Y and each of $G[X \cup \{a, b\}]$ and $G[Y \cup \{a, b\}]$ is not a path from a to b.

The 2-join was first defined by Cornuéjols and Cunningham [19]. A *2-join* in a graph G is a partition (X_1, X_2) of $V(G)$ with specified sets (A_1, A_2, B_1, B_2) such that the followings hold:

- $|X_1|, |X_2| \geq 3$.
- For $i = 1, 2$, $A_i \cup B_i \subseteq X_i$ and A_i, B_i are non-empty and disjoint.
- A_1 is complete to A_2, B_1 is complte to B_2 and these are the only adjacencies between X_1 and X_2.
- For $i = 1, 2$, $G[X_i]$ contains a path with one end in A_i and the other in B_i. Furthermore, $G[X_i]$ is not a path.

In this case, we call $(X_1, X_2, A_1, B_1, A_2, B_2)$ a *split* of (X_1, X_2). We also denote by C_i the set $X_i \setminus (A_i \cup B_i)$ for $i = 1, 2$. Note that the first three conditions of a 2-join are standard, while the last one is not. This is often referred to as *connected non-path 2-join* in literature. This condition varies since it depends on how we want to build our *blocks of decompositions* with respect to a 2-join. Here we show how to do it with respect to our definition. The *blocks of decompositions* of G with respect to (X_1, X_2) are the two graphs G_1, G_2 built as follows. We obtain G_1 by replacing X_2 by a *marker path* P_2 of length $k_2 \geq 3$, from a vertex a_2 complete to A_1, to a vertex b_2 complete to B_1 (the interior of P_2 has no neighbor in X_1). The block G_2 is obtained similarly by replacing X_1 by a marker path P_1 of length $k_1 \geq 3$ with two ends a_1, b_1. Note that the way we choose P_1, P_2 (as well as their lengths) depends on the class we are working with. This definition of a 2-join and this way of constructing blocks of decomposition with respect to a 2-join are consistent throughout this thesis (mainly for Chapter 3).

The next graph operation we introduce is called *substitution*. Given two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ (V_1 and V_2 are disjoint), and $v \in V_1$, we say that G is *obtained from G_1 by substituting G_2 for v*, or G is *obtained from G_1 by substituting v by G_2* if:

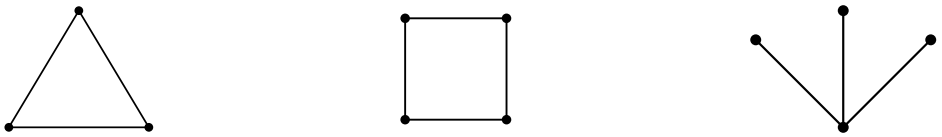
- $V(G) = V_1 \cup V_2 \setminus \{v\}$.
- $G[V_2] = G_2$.
- $G[V_1 \setminus \{v\}] = G_1[V_1 \setminus \{v\}]$.
- For every $v_1 \in V_1 \setminus \{v\}$ and $v_2 \in V_2$, $v_1v_2 \in E(G)$ if and only if $v_1v \in E(G_1)$.

Now we present another graph operation named *identification*. Given two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ (V_1 and V_2 are disjoint) and for some $1 \leq k \leq \min\{|V_1|, |V_2|\}$, let $X = \{x_1, \ldots, x_k\} \subseteq V_1$ and $Y = \{y_1, \ldots, y_k\} \subseteq V_2$ be such that $x_i x_j \in E_1$ if and only if $y_i y_j \in E_2$ for every $i, j \in \{1, 2, \ldots, k\}$. We say that G is *obtained from G_1 and G_2 by identifying x_i with y_i for every $i \in \{1, \ldots, k\}$* if:

- $V(G) = (V_1 \setminus X) \cup (V_2 \setminus Y) \cup Z$, where $Z = \{z_1, \ldots, z_k\}$.
- $G[V_1 \setminus X] = G_1[V_1 \setminus X]$.
- $G[V_2 \setminus Y] = G_2[V_2 \setminus Y]$.
- For every $i, j \in \{1, 2, \ldots, k\}$, $z_i z_j \in E(G)$ if and only if $x_i x_j \in E_1$ and $y_i y_j \in E_2$.
- For every $i \in \{1, \ldots, k\}$:
  - For every $u \in V_1 \setminus X$, $z_i u \in E(G)$ if and only if $x_i u \in E_1$.
  - For every $u \in V_2 \setminus Y$, $z_i u \in E(G)$ if and only if $y_i u \in E_2$.
- There is no edge in G between $V_1 \setminus X$ and $V_2 \setminus Y$.

1.4 Some particular graphs

In this section, some particular graphs are presented. Recall that a *triangle* is a graph isomorphic to a K_3. A *square* is a graph isomorphic to a C_4. A *claw* is a graph with vertex set $\{u, x, y, z\}$ and edge set $\{ux, uy, uz\}$, vertex u is called the *center* of that claw (see Figure 1.4).

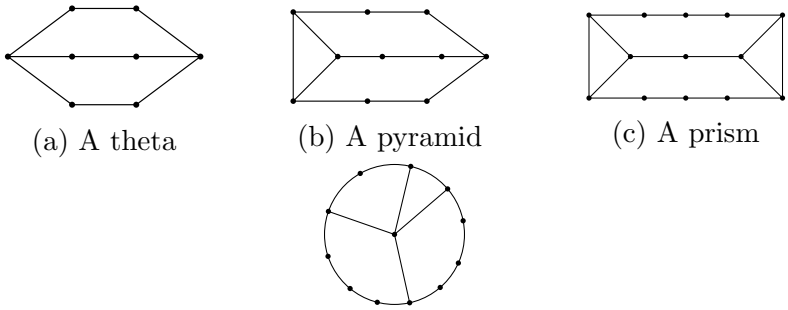


(a) triangle (b) square (c) claw

Figure 1.4: Some small graphs

Let us introduce Truemper configurations. These configurations play an important role in understanding the structure of several classes of objects, such as regular matroids, balanceable matrices and perfect graphs. In decomposition theorem, Truemper configurations appear both as excluded structures that are convenient to

work with, and as structures where the actual decomposition takes place. We refer the reader to [51] for a survey on Truemper configurations by Vušković. These configurations include: theta, pyramid, prism and wheel.



(a) A theta (b) A pyramid (c) A prism

(d) A wheel

Figure 1.5: Example of Truemper configurations

- Let x, y be two distinct vertices. A *theta* is a graph induced by three paths from x to y such that any two of them induce a hole (see Figure 1.5a). Note that all these three path are of length greater than 1 by this condition.

- Let x_1, x_2, x_3, y be four distinct vertices such that $\{x_1, x_2, x_3\}$ induces a triangle. A *pyramid* is a graph induced by three paths: P_1 from x_1 to y, P_2 from x_2 to y, P_3 from x_3 to y such that any two of them induce a hole (see Figure 1.5b). A pyramid is *long* if all three paths are of length greater than 1.

- Let $x_1, x_2, x_3, y_1, y_2, y_3$ be six distinct vertices such that $\{x_1, x_2, x_3\}$ and $\{y_1, y_2, y_3\}$ induce two triangles. A *prism* is a graph induced by three paths: P_1 from x_1 to y_1, P_2 from x_2 to y_2, P_3 from x_3 to y_3 such that any two of them induce a hole (see Figure 1.5c). A prism is *long* if all three paths are of length greater than 1.

- A *wheel* is a graph consisting of a hole H and a vertex $x \notin V(H)$ that has at least three neighbors on H (see Figure 1.5d).

Chapter 2

IKS4-free graphs

An *ISK4* is a subdivision of K_4 (K_4 is also an ISK4). A graph is *ISK4-free* if it does not contain any subdivision of K_4 as an induced subgraph. Series-parallel graphs and line graph of cubic graphs are some examples of ISK4-free graphs (see [38]). The class of ISK4-free graphs has recently been studied. In [38], a decomposition theorem for this class is given. However, it does not lead to a recognition algorithm. Since the class of ISK4-free graphs contains the line graph of every cubic graph, where finding the edge chromatic number is known to be NP-hard [32], we know that finding the chromatic number of ISK4-free graphs is also NP-hard. In Section 2.1, we propose new upper bound for the chromatic number of ISK4-free graphs and {ISK4, triangle}-free graphs. In Section 2.2, we give a polynomial-time algorithm for recognizing this class, which answers a question by Chudnovsky *et al.* [15] and Lévêque *et al.* [37]. We also note that the complexity of finding a maximum stable set in ISK4-free graphs remains open. The results of this chapter are covered in the following papers:

[I] N.K. Le. Chromatic number of ISK4-free graphs, *Graphs and Combinatorics*, 33(6): 1635-1646, 2017.

[II] N.K. Le. Detecting an induced subdivision of K_4, *arXiv preprint*, submitted. `https://arxiv.org/abs/1703.04637`

2.1 Chromatic number of ISK4-free graphs

2.1.1 Introduction

We recall the question concerning χ-boundedness: which induced subgraphs need to be forbidden to get a χ-bounded class of graphs? One way to forbid induced

structures is the following: fix a graph H, and forbid every induced subdivision of H. We denote by $Forb^*(H)$ the class of graphs that does not contain any subdivision of H as an induced subgraph. The class Forb$^*(H)$ has been proved to be χ-bounded for a number of graph H. Scott [45] proved that for any forest F, Forb$^*(F)$ is χ-bounded. In the same paper, he conjectured that Forb$^*(H)$ is χ-bounded for any graph H. Unfortunately, this conjecture has been disproved (see [42]). However, there is no general conjecture on which graph H, Forb$^*(H)$ is χ-bounded. This question is discussed in [9]. We focus on the question when $H = K_4$. In this case, Forb$^*(K_4)$ is the class of ISK4-free graphs. Since K_4 is forbidden, proving that the class of ISK4-free graphs is χ-bounded is equivalent to proving that there exists a constant c such that for every ISK4-free graph G, $\chi(G) \leq c$. Remark that the existence of such constant was pointed out in [38] as a consequence of a result in [36], but it is rather large ($\geq 2^{2^{2^{25}}}$) and very far from these two conjectures:

Conjecture 2.1.1 (Lévêque, Maffray, Trotignon 2012 [38]). *If G is an ISK4-free graph, then $\chi(G) \leq 4$.*

Conjecture 2.1.2 (Trotignon, Vušković 2016 [49]). *If G is an {ISK4, triangle}-free graph, then $\chi(G) \leq 3$.*

No better upper bound is known even for the chromatic number of {ISK4, triangle}-free graphs. However, attempts were made toward these two conjectures. The optimal bound is known for the chromatic number of {ISK4, wheel}-free graphs and {ISK4, triangle, C_4}-free graphs:

Theorem 2.1.3 (Lévêque, Maffray, Trotignon 2012 [38]). *Every {ISK4, wheel}-free graph is 3-colorable.*

Theorem 2.1.4 (Trotignon, Vušković 2016 [49]). *Every ISK4-free graph of girth at least 5 contains a vertex of degree at most 2 and is 3-colorable.*

The proof of Theorems 2.1.3 and 2.1.4 relies on structural decompositions. One way to prove Conjectures 2.1.1 and 2.1.2 is to find a vertex of small degree. This approach is successfully used in [49] to prove Theorem 2.1.4. Two following conjectures will imply the correctness of Conjectures 2.1.1 and 2.1.2 (as shown in [38]):

Conjecture 2.1.5 (Trotignon 2015 [47]). *Every {ISK4, $K_{3,3}$, prism, $K_{2,2,2}$}-free graph contains a vertex of degree at most three.*

Conjecture 2.1.6 (Trotignon, Vušković 2016 [49]). *Every {ISK4, $K_{3,3}$, triangle}-free graph contains a vertex of degree at most two.*

However, we find a new bound for the chromatic number of ISK4-free graphs using another approach. Our main results are the following theorems:

Theorem 2.1.7. *Let G be an $\{$ISK4, triangle$\}$-free graph. Then $\chi(G) \leq 4$.*

Theorem 2.1.8. *Let G be an ISK4-free graph. Then $\chi(G) \leq 24$.*

Remark that the bounds we found are much closer to the bound of the conjectures than the known ones. The main tool that we use to prove these theorems is classical. It is often used to prove χ-boundedness results relying on the layers of neighborhood. This section is organized as follows. We first introduce some notations in Section 2.1.2. Sections 2.1.3 and 2.1.4 are devoted to the proof of Theorem 2.1.7 and 2.1.8, respectively.

2.1.2 Preliminaries

In this section, we present some notations and useful lemmas which will be used later in our proof. Let $G = (V, E)$ be a graph, we denote by $|G|$ the number of its vertices. If the context is clear, we will sometimes write G for $V(G)$. A complete bipartite or tripartite graph is *thick* if it contains a $K_{3,3}$. Let $S = \{u_1, u_2, u_3, u_4\}$ induces a square (i.e. C_4) in G with u_1, u_2, u_3, u_4 in this order along the square. A *link* of S is a path P of G with ends p, p' such that either $p = p'$ and $N_S(p) = S$, or $N_S(p) = \{u_1, u_2\}$ and $N_S(p') = \{u_3, u_4\}$, or $N_S(p) = \{u_1, u_4\}$ and $N_S(p') = \{u_2, u_3\}$, and no interior vertex of P has a neighbor in S. A *rich square* is a graph K that contains a square S as an induced subgraph such that $K \setminus S$ has at least two components and every component of $K \setminus S$ is a link of S. For example, $K_{2,2,2}$ is a rich square (it is the smallest one). We refer the reader to Chapter 1 for the remaining notions (clique cutset, proper 2-cutset, prism, ...).

We use in this section some decomposition theorems from [38]:

Lemma 2.1.9 (see Lemma 3.3 in [38])**.** *Let G be an ISK4-free graph that contains $K_{3,3}$. Then either G is a thick complete bipartite or complete tripartite graph, or G has a clique cutset of size at most 3.*

Lemma 2.1.10 (see Lemmas 6.1 and 7.2 in [38])**.** *Let G be an ISK4-free graph that contains a rich square or a prism. Then either G is the line graph of a graph with maximum degree 3, or G is a rich square, or G has a clique cutset of size at most 3 or G has a proper 2-cutset.*

Reducing a flat path P of length at least 2 means deleting its interior and adding an edge between its two ends. The following lemma shows that a graph remains ISK4-free after reducing a flat path:

Lemma 2.1.11 (see Lemma 11.1 in [38]). *Let G be an ISK4-free graph. Let P be a flat path of length at least 2 in G and G' be the graph obtained from G by reducing P. Then G' is ISK4-free.*

It is shown in [38] that clique cutsets and proper 2-cutsets are useful for proving Conjecture 2.1.1 in the inductive sense. If we can find such a cutset in G, then we immediately have a bound for the chromatic number of G, since $\chi(G) \leq \max\{\chi(G_1), \chi(G_2)\}$, where G_1 and G_2 are two blocks of decomposition of G with respect to that cutset (see the proof of Theorem 1.4 in [38]). Therefore, we only have to prove Conjecture 2.1.1 for the class of {ISK4, $K_{3,3}$, prism, $K_{2,2,2}$}-free graphs and prove Conjecture 2.1.2 for the class of {ISK4, $K_{3,3}$, triangle}-free graphs since the existence of $K_{3,3}$, prism or $K_{2,2,2}$ implies a good cutset or a structure that can be appropriately colored by Lemmas 2.1.9 and 2.1.10.

For $S, C \subseteq V(G)$, we say that S *dominates* C if $N_C(S) = C$. The *distance* between two vertices x, y in $V(G)$ is the length of a shortest path from x to y in G. Let $u \in V(G)$ and i be an integer, we denote by $N_i(u)$ the set of vertices of G that are of distance exactly i from u. Note that there are no edges between $N_i(u)$ and $N_j(u)$ for every i, j such that $|i - j| \geq 2$.

Lemma 2.1.12. *Let G be a graph, $u \in V(G)$ and i be an integer ≥ 1. Let x, y be two distinct vertices in $N_i(u)$. Then, there exists a path P in G from x to y such that $V(P) \subseteq \{u\} \cup N_1(u) \cup \ldots \cup N_i(u)$ and $|V(P) \cap N_j(u)| \leq 2$ for every $j \in \{1, \ldots, i\}$.*

Proof. We prove this by induction on i. If $i = 1$, we have $x, y \in N_1(u)$. If $xy \in E(G)$, we choose $P = xy$, otherwise, choose $P = xuy$. Suppose that the lemma is true until $i = k$, we prove that it is also true for $i = k+1$. If $xy \in E(G)$, we choose $P = xy$. If x and y have a common neighbor x' in $N_k(u)$, then we choose $P = xx'y$. Otherwise, let x', y' be the vertices in $N_k(u)$ such that $x'x, y'y \in E(G)$. Note that $xy', x'y \notin E(G)$. We choose $P = P' \cup \{x, y\}$, where P' is the path with two ends x' and y' generated by applying induction hypothesis. □

Such a path P in Lemma 2.1.12 is called an *upstairs path* of $\{x, y\}$. For three distinct vertices $x, y, z \in V(G)$, a graph H is a *confluence* of $\{x, y, z\}$ if it is one of the two following types:

- Type 1:
  - $V(H) = V(P_x) \cup V(P_y) \cup V(P_z)$.
  - P_x, P_y, P_z are three paths having a common end u and $P_x \setminus u$, $P_y \setminus u$, $P_z \setminus u$ are pairwise disjoint. The other ends of P_x, P_y, P_z are x, y, z, respectively.

  - These are the only edges in H.

- Type 2:
  - $V(H) = V(P_x) \cup V(P_y) \cup V(P_z)$.
  - P_x is a path with ends x and x'.
  - P_y is a path with ends y and y'.
  - P_z is a path with ends z and z'.
  - P_x, P_y, P_z are pairwise disjoint.
  - $x'y'z'$ is a triangle.
  - These are the only edges in H.

If H is a confluence of Type 1, the vertex u is called the *center* of H and if H is a confluence of Type 2, the triangle $x'y'z'$ is called the *center triangle* of H. Note that the length of P_x can be 0 when $x = u$ (for Type 1) or $x = x'$ (for Type 2).

Lemma 2.1.13. *Let G be a graph, $u \in V(G)$ and i be an integer ≥ 1. Let x, y, z be three distinct vertices in $N_i(u)$. Then, there exists a set $S \subseteq \{u\} \cup N_1(u) \cup \ldots \cup N_{i-1}(u)$ such that $G[S \cup \{x, y, z\}]$ is a confluence of $\{x, y, z\}$.*

Proof. Let G' be the subgraph of G induced by $\{u\} \cup N_1(u) \cup \ldots \cup N_{i-1}(u) \cup \{x, y, z\}$. It is clear that G' is connected. Let P be a path in G' from x to y and Q be a path in G' from z to P (one end of Q is in P). We choose P and Q subject to minimality of $|V(P \cup Q)|$. It is easy to see that $G[V(P \cup Q)]$ is a confluence of $\{x, y, z\}$. □

The notions of upstairs path and confluence are very useful to find induced structures in our graph since they establish a way to connect two or three vertices of the same layer through only the upper layers.

Lemma 2.1.14. *Let G be a graph and $u \in V(G)$. Then:*

$$\chi(G) \leq \max_{i \text{ odd}} \chi(G[N_i(u)]) + \max_{j \text{ even}} \chi(G[N_j(u)]).$$

Proof. It is clear that in G, there are no edges between $N_i(u)$ and $N_j(u)$ if $i \neq j$ and i, j are of the same parity. Therefore, we can color all the odd layers with $\max_{i \text{ odd}} \chi(G[N_i(u)])$ colors and all the even layers with $\max_{j \text{ even}} \chi(G[N_j(u)])$ other colors. The lemma follows. □

2.1.3 Proof of Theorem 2.1.7

The next lemma shows that if there is a set S that dominates some hole C, then there must exist some vertices in S which have very few (one or two) neighbors in C.

Lemma 2.1.15. *Let G be an {ISK4, triangle, $K_{3,3}$}-free graph and C be a hole in G. Let $S \subseteq V(G) \setminus C$ such that S dominates C. Then one of the following cases holds:*

1. *There exist four distinct vertices u_1, u_2, u_3, u_4 in S and four distinct vertices v_1, v_2, v_3, v_4 in C such that for $i \in \{1,2,3,4\}$, $N_C(u_i) = \{v_i\}$.*

2. *There exist three distinct vertices u_1, u_2, u_3 in S and three distinct vertices v_1, v_2, v_3 in C such that for $i \in \{1,2,3\}$, $N_C(u_i) = \{v_i\}$ and v_1, v_2, v_3 are pairwise non-adjacent.*

3. *There exist three distinct vertices u_1, u_2, u_3 in S and four distinct vertices v_1, v_2, v_3, v_3' in C such that $N_C(u_1) = \{v_1\}$, $N_C(u_2) = \{v_2\}$, $N_C(u_3) = \{v_3, v_3'\}$ and v_1, v_3, v_2, v_3' appear in this order along C.*

Proof. We prove Lemma 2.1.15 by induction on the length of hole C. First, suppose that the length of C is 4 and $C = c_0c_1c_2c_3c_0$. Since G is triangle-free, a vertex in S can only have one or two neighbors in C. We consider two cases:

- If some vertex $u \in S$ has two neighbors in C, w.l.o.g, suppose $N_C(u) = \{c_0, c_2\}$. Since S dominates C, there exists some vertices v, $w \in S$ such that $vc_1, wc_3 \in E$. If $v = w$ then $\{u, v\} \cup C$ induces $K_{3,3}$ (if $uv \in E$) or an ISK4 (if $uv \notin E$), a contradiction. Therefore, $v \neq w$ and u, v, w are three vertices satisfying output 3 of the lemma.

- If every vertex in S has exactly one neighbor in C, output 1 of the lemma holds.

Now, we may assume that $|C| \geq 5$ and the lemma is true for every hole of length at most $|C| - 1$. A vertex $u \in S$ is a *bivertex* if $N_C(u) = \{u', u''\}$ and the two paths P_1, P_2 from u' to u'' in C are of lengths at least 3. Suppose that S contains such a bivertex u. Let $C_1 = P_1 \cup \{u\}$, $C_2 = P_2 \cup \{u\}$, note that $|C_1|, |C_2| < |C|$. Consider the graph G' obtained from G as follows: $V(G') = V(G) \cup \{a, b, c\}$, $E(G') = E(G) \cup \{au, bu', cu''\}$. It is clear that G' is {ISK4, triangle, $K_{3,3}$}-free. Let $S_1 = \{v \in S \setminus u | N_{C_1}(v) \neq \emptyset\} \cup \{a, b, c\}$ and $S_2 = \{v \in S \setminus u | N_{C_2}(v) \neq \emptyset\} \cup \{a, b, c\}$. By applying the induction hypothesis on S_1 and C_1, we obtain that there is some

vertex $x \in S$ such that x has exactly one neighbor in P_1 and this neighbor is in P_1^* (x can be adjacent to u). We claim that x has exactly one neighbor in C. Indeed, if x has exactly one neighbor x' in P_2^* then $C \cup \{x, u\}$ induces an ISK4 (if $xu \notin E(G)$) or $C_1 \cup \{x\} \cup Q$ induces an ISK4 (if $xu \in E(G)$), where Q is the shorter path in one of the two paths in C: $x'P_2u'$ and $x'P_2u''$, a contradiction. If x has at least two neighbors in P_2^*, let x', x'' be the neighbors of x closest to u', u'' on P_2^*, respectively. Then $C_1 \cup \{x\} \cup x'P_2u' \cup x''P_2u''$ induces an ISK4 (if $xu \notin E(G)$) or $C_1 \cup \{x\} \cup x'P_2u'$ induces an ISK4 (if $xu \in E(G)$), a contradiction. So, x has no neighbor in P_2^* and has exactly one neighbor in C as claimed. Similarly, by applying the induction hypothesis on S_2 and C_2, we know that there is some vertex $y \in S$ such that y has exactly one neighbor in P_2^* and this is also its only neighbor in C. Now, $\{x, y, u\}$ satisfies output 3 of the lemma. Hence, we may assume that S contains no bivertex.

Note that since G is ISK4-free, no vertex in S has exactly three neighbors in C. Suppose that there is some vertex u in S which has at least four neighbors in C. Let $N_C(u) = \{u_0, \ldots, u_k\}$ where $u_0, \ldots, u_k$ ($k \geq 3$) appear in that order along C. Let $P_u(i, i+3)$ be the path of C from u_i to u_{i+3} which contains u_{i+1} and u_{i+2} and define $\text{amp}(u, C) = \max_{i=0}^{k} |P_u(i, i+3)|$ (the index is taken modulo $k+1$). Note that this notion is defined only for a vertex with at least four neighbors in C. Let $v \in S$ be such that $\text{amp}(v, C)$ is maximum. Similarly, let $N_C(v) = \{v_0, \ldots, v_k\}$ where $v_0, \ldots, v_k$ ($k \geq 3$) appear in that order along C. W.l.o.g suppose that $P_v(0, 3)$ is the longest path among all paths of the form $P_v(i, i+3)$. Let P_0, P_1, P_2 be the subpaths of $P_v(0, 3)$ from v_0 to v_1, v_1 to v_2, v_2 to v_3, respectively. Let $C_0 = \{v\} \cup P_0$, $C_1 = \{v\} \cup P_1$ and $C_2 = \{v\} \cup P_2$. Consider the graph G' obtained from G as follows: $V(G') = V(G) \cup \{a, b, c\}$, $E(G') = E(G) \cup \{av, bv_0, cv_1\}$. It is clear that G' is {ISK4, triangle, $K_{3,3}$}-free. Let $S_0 = \{u \in S \setminus v | N_{C_0}(u) \neq \emptyset\} \cup \{a, b, c\}$. By applying the induction hypothesis on S_0 and C_0, we obtain that there is some vertex $x \in S$ such that x has exactly one neighbor x_0 in P_0 and x_0 is in P_0^* (x can be adjacent to v). We claim that x has exactly one neighbor in C. Suppose that x has some neighbor in P_1. Let x_1, x_2 be the neighbors of x in P_1 which are closest to v_1 and v_2, respectively (x_1 and x_2 could be equal). Then we have $\{x, v\} \cup P_0 \cup v_1P_1x_1 \cup v_2P_1x_2$ induces an ISK4 (if $xv \notin E(G)$) or $\{x, v\} \cup P_0 \cup v_1P_1x_1$ induces an ISK4 (if $xv \in E(G)$), a contradiction. Therefore, x has no neighbor in P_1. Suppose that x has some neighbor in P_2, let x_1 be the neighbor of x in P_2 which is closest to v_2. Let Q be the path from x_0 to x_1 in C which contains v_1. We have $\{x, v\} \cup Q \cup v_0P_0x_0$ induces an ISK4 (if $xv \notin E(G)$) or $\{x, v\} \cup Q$ induces an ISK4 (if $xv \in E(G)$), a contradiction. Hence, x has no neighbor in P_2. Now if x has at least four neighbors in C, $\text{amp}(x, C) > \text{amp}(v, C)$, a contradiction to the choice of v. Hence, x can have at most one neighbor in the path from v_0 to v_3 in C which does not contain v_1.

Suppose x has one neighbor x' in that path. By the assumption that we have no bivertex, $x'v_0, v_0x_0 \in E(G)$. Let Q be the path from v_k to x' in C which does not contain v_0. We have $\{x, x', v_0, x_0, v\} \cup Q \cup v_1P_0x_0$ induces an ISK4 (if $xv \notin E(G)$) or $\{x, x', v_0, x_0, v\} \cup Q$ induces an ISK4 (if $xv \in E(G)$), a contradiction. Hence, x_0 is the only neighbor of x in C, as claimed. Similarly, we can prove that there exist two vertices $y, z \in S$ such that they have exactly one neighbor in C which are in P_1^*, P_2^*, respectively. Note that the proof for y is not formally symmetric to the one for x and z, but the proof is actually the same. In particular, a vertex y with a unique neighbor in P_1^*, no neighbor in P_0, P_2 and at least four neighbors in C also yields a contradiction to the maximality of $\text{amp}(v, C)$. Therefore, $\{x, y, z\}$ satisfies output 2 of the lemma. Now, we can assume that no vertex in S has at least four neighbors in C.

Hence, every vertex in S either has exactly one neighbor in C or exactly two neighbors in C and is not a bivertex. Suppose there is some vertex u that has two neighbors u', u'' on C and let $x \in C$ be such that $xu', xu'' \in E$. Let $v \in S$ be a vertex adjacent to x. If v has another neighbor x' in C then x' must be adjacent to u' or u'', since v is not a bivertex. So, we have that $\{u, v, x', u', x, u''\}$ induces an ISK4 (if $uv \in E(G)$) or $\{u, v\} \cup C$ induces an ISK4 (if $uv \notin E(G)$), a contradiction. So, v has only one neighbor x in C. By the same argument, some vertex in $V(C) \setminus \{u', x, u''\}$ is the unique neighbor in C of some vertex in S. Hence, if we have at least one vertex which has two neighbors on C, then output 3 holds. If every vertex has exactly one neighbor in C, the output 1 holds, which completes the proof. □

Lemma 2.1.16. *Let G be an $\{$ISK4, triangle, $K_{3,3}\}$-free graph and $u \in V(G)$. For every $i \geq 1$, $G[N_i(u)]$ does not contain any hole.*

Proof. Suppose for some i, $G[N_i(u)]$ contains a hole C. For every vertex $v \in C$, there exists a vertex $v' \in N_{i-1}(u)$ such that $vv' \in E$. Hence there exists a subset $S \subseteq N_{i-1}(u)$ such that S dominates C. Let us apply Lemma 2.1.15 for S and C:

- If output 1 or 2 of Lemma 2.1.15 holds, then there exist three distinct vertices u_1, u_2, u_3 in S and three distinct vertices v_1, v_2, v_3 in C such that for $i \in \{1, 2, 3\}$, $N_C(u_i) = \{v_i\}$. By Lemma 2.1.13, since G is triangle-free, there exists a confluence F of $\{u_1, u_2, u_3\}$ of Type 1, so $F \cup C$ induces an ISK4, a contradiction.

- If output 3 of Lemma 2.1.15 holds, then there exist two distinct vertices u_1, u_2 in S and three distinct vertices v_1, v_2, v_2' in C such that $N_C(u_1) = v_1$, $N_C(u_2) = \{v_2, v_2'\}$. By Lemma 2.1.12, there exists an upstairs path P of $\{u_1, u_2\}$, so $P \cup C$ induces an ISK4, a contradiction.

□

Proof of Theorem 2.1.7 We prove the theorem by induction on the number of vertices of G. Suppose that G has a clique cutset K. So $G \setminus K$ can be partitioned into two sets X, Y such that there is no edge between them. By induction hypothesis $\chi(G[X \cup K]) \leq 4$ and $\chi(G[Y \cup K]) \leq 4$, therefore $\chi(G) = \max\{\chi(G[X \cup K]), \chi(G[Y \cup K])\} \leq 4$. Hence we may assume that G has no clique cutset. If G contains a $K_{3,3}$, then by Lemma 2.1.9, G is a thick complete bipartite graph and $\chi(G) \leq 2$. So we may assume that G contains no $K_{3,3}$. By Lemma 2.1.16, for every $u \in V(G)$, for every $i \geq 1$, $G[N_i(u)]$ is a forest, hence $\chi(G[N_i(u)]) \leq 2$. By Lemma 2.1.14, $\chi(G) \leq 4$, which completes the proof. □

2.1.4 Proof of Theorem 2.1.8

A *boat* is a graph consisting of a hole C and a vertex v that has exactly four consecutive neighbors in C ($N_C(v)$ induces a C_4 if $|C| = 4$ or a P_4 if $|C| \geq 5$). A *4-wheel* is a particular boat whose hole is of length 4. Let $\mathcal{C}_1$ be the class of {ISK4, $K_{3,3}$, prism, boat}-free graphs, $\mathcal{C}_2$ be the class of {ISK4, $K_{3,3}$, prism, 4-wheel}-free graphs and $\mathcal{C}_3$ be the class of {ISK4, $K_{3,3}$, prism, $K_{2,2,2}$}-free graphs. Remark that $\mathcal{C}_1 \subsetneq \mathcal{C}_2 \subsetneq \mathcal{C}_3 \subsetneq$ ISK4-free graphs.

Lemma 2.1.17. *Let G be a graph in $\mathcal{C}_1$. Then $\chi(G) \leq 6$.*

Proof. We prove first the following.

Claim 2.1.18. *Let $u \in V(G)$ and $i \geq 1$. Then $G[N_i(u)]$ contains no triangle and no C_4.*

Proof. Suppose $G[N_i(u)]$ contains a triangle abc. No vertex is complete to abc since G is K_4-free. Suppose that there is some vertex $x \in N_{i-1}(u)$ which has exactly two neighbors in the triangle, w.l.o.g. assume that they are a and b. Let y be some vertex in $N_{i-1}(u)$ adjacent to c and P be an upstairs path of $\{x, y\}$. If y has exactly one neighbor in abc (which is c), then $P \cup \{a, b, c\}$ induces an ISK4, a contradiction. Hence y must have another neighbor in C, say a up to symmetry. In this case, $P \cup \{a, b, c\}$ induces a boat, a contradiction. Then every vertex in $N_{i-1}(u)$ has exactly one neighbor in abc. Suppose there are three vertices $x, y, z \in N_{i-1}(u)$ such that $N_{abc}(x) = \{a\}$, $N_{abc}(y) = \{b\}$ and $N_{abc}(z) = \{c\}$. By Lemma 2.1.13, there exists a confluence S of $\{x, y, z\}$. If S is of Type 1, then $S \cup \{a, b, c\}$ induces an ISK4, a contradiction. If S is of Type 2, then $S \cup \{a, b, c\}$ induces a prism, a contradiction. Hence, $G[N_i(u)]$ contains no triangle.

Suppose $N_i(u)$ contains a C_4, namely $abcd$. Every vertex can only have zero, one or two neighbors in $abcd$ since a 4-wheel is a boat. Suppose there is some vertex $x \in N_{i-1}(u)$ which has exactly two non-adjacent neighbors in $\{a, b, c, d\}$, say $N_{abcd}(x) = \{a, c\}$. Let y be some vertex in $N_{i-1}(u)$ adjacent to d and P be an upstairs path of $\{x, y\}$. If $yb \in E$, then $\{x, y, a, b, c, d\}$ induces an ISK4 (if $xy \notin E$) or a $K_{3,3}$ (if $xy \in E$), a contradiction. If $ya \in E$, $P \cup \{a, c, d\}$ induces an ISK4, a contradiction. Then $yc \notin E$ also by symmetry, and $N_{abcd}(y) = \{d\}$. In this case $P \cup \{a, b, c, d\}$ induces an ISK4, a contradiction. Therefore, no vertex in $N_{i-1}(u)$ has two non-adjacent neighbors in $\{a, b, c, d\}$. Now, suppose that there is some vertex $x \in N_{i-1}(u)$ which has exactly two consecutive neighbors $\{a, b\}$ in $abcd$. Let y be some vertex in $N_{i-1}(u)$ adjacent to d and P be an upstairs path of $\{x, y\}$. If y is adjacent to c, then $P \cup \{a, b, c, d\}$ induces a prism, a contradiction. If $N_{abcd}(y) = \{d\}$, then $P \cup \{a, b, c, d\}$ induces an ISK4, a contradiction. Hence $N_{abcd}(y) = \{a, d\}$. Let z be some vertex in $N_{i-1}(u)$ adjacent to c, P_{xz} be an upstairs path of $\{x, z\}$ and P_{yz} be an upstairs path of $\{y, z\}$. If $zb \in E$, $P_{yz} \cup \{a, b, c, d\}$ induces a prism, a contradiction. If $zd \in E$, $P_{xz} \cup \{a, b, c, d\}$ induces a prism, a contradiction. Hence $N_{abcd}(z) = \{c\}$. In this case, $P_{xz} \cup \{a, b, c, d\}$ induces an ISK4, a contradiction. Therefore, there is no vertex in $N_{i-1}(u)$ having two neighbors in $\{a, b, c, d\}$. So, there are three vertices $x, y, z \in N_{i-1}(u)$ such that $N_{abcd}(x) = \{a\}$, $N_{abcd}(y) = \{b\}$, $N_{abcd}(z) = \{c\}$. By Lemma 2.1.13, there exists a confluence S of $\{x, y, z\}$. If S is of Type 1, $S \cup \{a, b, c, d\}$ induces an ISK4, a contradiction. If S is of Type 2, $S \cup \{a, b, c\}$ induces an ISK4, a contradiction. Therfore, $G[N_i(u)]$ contains no C_4. □

By Claim 2.1.18, the girth of $N_i(u)$ is at least 5 for $i \geq 1$. By Theorem 2.1.4, $\chi(G[N_i(u)]) \leq 3$. By Lemma 2.1.14, $\chi(G) \leq 6$, which completes the proof. □

Lemma 2.1.19. *Let G be a graph in $\mathcal{C}_2$. Then $\chi(G) \leq 12$.*

Proof. We first prove that: for any $u \in V(G)$ and $i \geq 1$, $G[N_i(u)]$ contains no boat. We may assume that $i \geq 2$, since $G[N_1(u)]$ is triangle-free, the conclusion holds for $i = 1$. Suppose for contradiction that $G[N_i(u)]$ contains a boat consisting of a hole C and a vertex x that has four neighbors a, b, c, d in this order on C. Since G contains no 4-wheel, we can assume that $|C| \geq 5$ and $\{a, b, c, d\}$ induces a P_4. Let P be the path from a to d in C which does not go through b.

Claim 2.1.20. *No vertex in $N_{i-1}(u)$ is adjacent to both b and c.*

Proof. Suppose there is a vertex $y \in N_{i-1}(u)$ adjacent to both b and c. Since $\{x, y, b, c\}$ does not induce K_4, $xy \notin E$. If $ya \in E$, $\{a, b, c, x, y\}$ induces a 4-wheel, a contradiction. Hence, $ya \notin E$. We also have $yd \notin E$ by symmetry. We claim

that $N_C(y) = \{b, c\}$. Suppose that y has some neighbor in P^*. If y has exactly one neighbor in P^*, then $\{y\} \cup C$ induces an ISK4, a contradiction. If y has exactly two consecutive neighbors in P^*, then $C \cup \{x, y\} \setminus \{c\}$ induces a prism, a contradiction. If y has at least three neighbors in P^*, or two neighbors in P^* that are not consecutive, then let z be the one closest to a and t be the one closest to d. Then $\{x, y, b\} \cup zPa \cup tPd$ induces an ISK4, a contradiction. So $N_C(y) = \{b, c\}$. Let z be a vertex in $N_{i-1}(u)$ which has a neighbor in P^* and P_{yz} be an upstairs path of $\{y, z\}$. If z has exactly one neighbor in C, then $P_{yz} \cup C$ induces an ISK4, a contradiction. If z has exactly two consecutive neighbors in C, then $P_{yz} \cup C$ induces a prism, a contradiction. If z has at least three neighbors in C or two neighbors in C which are not consecutive, let t, w be the ones closest to b, c in C, respectively. Let Q be the path from t to w in C which contains b. We have that $P_{yz} \cup Q$ induces an ISK4, a contradiction. □

By Claim 2.1.20, let y, z be two distinct vertices in $N_{i-1}(u)$ such that $yb, zc \in E$, $yc, zb \notin E$ and P_{yz} be an upstairs path of $\{y, z\}$.

Claim 2.1.21. *$xy, xz \in E$.*

Proof. Suppose $xy \notin E$. Then $xz \notin E$, otherwise $P_{yz} \cup \{x, b, c\}$ induces an ISK4. Let $t \in N_{i-1}(u)$ such that $tx \in E$, let P_{ty} and P_{tz} be upstairs paths of $\{t, y\}$ and $\{t, z\}$, respectively. If $tb \in E$, then $tc \notin E$ by Claim 2.1.20, and $P_{tz} \cup \{x, b, c\}$ induces an ISK4, a contradiction. If $tc \in E$, then $tb \notin E$ by Claim 2.1.20 and $P_{ty} \cup \{x, b, c\}$ induces an ISK4, a contradiction. So $N_{xbc}(t) = \{x\}$. By Lemma 2.1.13, let S be a confluence of $\{y, z, t\}$. If S is of Type 1, $S \cup \{x, b, c\}$ induces an ISK4, a contradiction. If S is of Type 2, $S \cup \{x, b, c\}$ induces a prism, a contradiction. Then $xy \in E$. Symmetrically, $xz \in E$. □

By Claim 2.1.21, $yz \notin E$ (for otherwise $\{x, b, c, y, z\}$ induces a 4-wheel).

Claim 2.1.22. *$N_C(y) = \{b\}$ and $N_C(z) = \{c\}$.*

Proof. We prove only $N_C(y) = \{b\}$, the other conclusion is proved similarly. First, $ya \notin E$, otherwise $\{y, x, a, b\}$ induces a K_4. We also have $yd \notin E$, otherwise $\{x, y, b, c, d\}$ induces a 4-wheel. If y has some neighbor in P^*, let t be the one closest to a. In this case, $tPa \cup \{x, y, b\}$ induces an ISK4, a contradiction. Hence $N_C(y) = \{b\}$. □

Let t be a vertex in $N_{i-1}(u)$ such that $ta \in E$ and P_{yt} be an upstairs path of $\{y, t\}$. By Claim 2.1.22, $tb, tc \notin E$. We have $tx \in E$, otherwise $P_{yt} \cup \{x, a, b\}$ induces an ISK4. Suppose that $N_C(t) = \{a\}$. There exists a confluence S of $\{t, y, z\}$ by

Lemma 2.1.13. If S is of Type 1, $S \cup C$ induces an ISK4, a contradiction. If S is of Type 2, $S \cup \{a, b, c\}$ induces an ISK4, a contradiction. Hence, t must have some neighbor in $P \setminus \{a\}$, let w be the one closest to d along P and P_w be the path from a to w in C which contains b. Note that $wa \notin E$, since otherwise $(V(C) \setminus \{b, c\}) \cup \{x, t\}$ induces an ISK4.

Claim 2.1.23. *t has some neighbor in P_{yz}.*

Proof. Suppose that t has no neighbor in P_{yz}. Because $G[u \cup N_1(u) \cup \ldots \cup N_{i-2}(u)]$ is connected, there exists a path Q from t to some t' such that $Q \setminus \{t\} \subseteq u \cup N_1(u) \cup \ldots \cup N_{i-2}(u)$ and t' is the only vertex in Q which has some neighbor in P_{yz}. If t' has exactly one neighbor in P_{yz}, then $P_w \cup Q \cup P_{yz}$ induces an ISK4, a contradiction. If t' has exactly two consecutive neighbors in P_{yz}, then $Q \cup P_{yz} \cup \{a, b, c\}$ induces an ISK4. If t' has at least three neighbors in P_{yz} or two neighbors in P_{yz} which are not consecutive, let y', z' be the one closest to y, z, respectively, then $Q \cup P_w \cup y'P_{yz}y \cup z'P_{yz}z$ induces an ISK4, a contradiction. Then t must have some neighbor in P_{yz}. □

Let $y', z' \in P_{yz}$ such that $y'y, z'z \in E$. Since $t \in N_{i-1}(u)$, $N_{P_{yz}}(t) \subseteq \{y, z, y', z'\}$. If t has exactly one neighbor in P_{yz}, then $\{t\} \cup P_{yz} \cup P_w$ induces an ISK4, a contradiction. If t has exactly two neighbors in P_{yz}, then $\{t, a, b, c\} \cup P_{yz}$ induces an ISK4, a contradiction. If t has exactly three neighbors in P_{yz}, then $\{t, b, c\} \cup P_{yz}$ induces an ISK4, a contradiction. Hence, t has four neighbors in P_{yz} or $N_{P_{yz}}(t) = \{y, z, y', z'\}$. In particular, $ty \in E$ and $\{x, t, y, a, b\}$ induces a 4-wheel, a contradiction. Hence, $G[N_i(u)]$ is boat-free.

Now, for every $i \geq 1$, $G[N_i(u)] \in \mathcal{C}_1$. By Lemma 2.1.17, $\chi(G[N_i(u)]) \leq 6$. By Lemma 2.1.14, $\chi(G) \leq 12$, completing the proof. □

Lemma 2.1.24. *Let G be a graph in $\mathcal{C}_3$. Then $\chi(G) \leq 24$.*

Proof. Let $u \in V(G)$ and $i \geq 1$. We claim that $G[N_i(u)]$ contains no 4-wheel. Suppose that $G[N_i(u)]$ contains a 4-wheel consisting of a hole $abcd$ and a vertex x complete to $abcd$. By similar argument as in the proof of Lemma 2.1.17 (the proof of C_4-free), the hole $abcd$ cannot be dominated by only the vertices in $N_{i-1}(u)$ which has one or two neighbors in $abcd$. Hence, there exists some vertex $y \in N_{i-1}(u)$ complete to $abcd$. It is clear that $xy \notin E$, otherwise $\{x, y, a, b\}$ induces a K_4. Now, $\{x, y, a, b, c, d\}$ induces a $K_{2,2,2}$, a contradiction. So, $G[N_i(u)]$ contains no 4-wheel. By Lemma 2.1.19, $\chi(G[N_i(u)]) \leq 12$. By Lemma 2.1.14, we have $\chi(G) \leq 24$, which proves the lemma. □

Before the main proof, we have several lemmas proving the bound of chromatic number of some basic graphs.

Lemma 2.1.25. *Let G be the line graph of a graph H with maximum degree three. Then $\chi(G) \leq 4$.*

Proof. To prove that G is 4-colorable, we only need to prove that H is 4-edge-colorable. But since the maximum degree of H is three, this is a direct consequence of Vizing's theorem (see [6]). □

Lemma 2.1.26. *Let G be a rich square. Then $\chi(G) \leq 4$.*

Proof. By the definition of a rich square, there is $S = \{u_1, u_2, u_3, u_4\}$ that induces a square in G such that every component of $G \setminus S$ is a link of S. Assume that u_1, u_2, u_3, u_4 appear in this order along the square. We show a 4-coloring of G as follows. Assign color 1 to $\{u_1, u_3\}$ and color 2 to $\{u_2, u_4\}$. Let P be a component of $G \setminus S$ with ends p, p'. If $p = p'$, give it color 3. If $p \neq p'$, give p, p' color 3, 4, respectively and assign color 1 and 2 alternately to the internal vertices of P. □

Proof of Theorem 2.1.8 We prove the theorem by induction on the number of vertices of G. Suppose that G has a clique cutset K. So $G \setminus K$ can be partitioned into two sets X, Y such that there are no edges between them. By the induction hypothesis, $\chi(G[X \cup K]) \leq 24$ and $\chi(G[Y \cup K]) \leq 24$, therefore $\chi(G) = \max\{\chi(G[X \cup K]), \chi(G[Y \cup K])\} \leq 24$. Hence we may assume that G has no clique cutset. If G contains a $K_{3,3}$, then by Lemma 2.1.9, G is a thick complete bipartite graph or complete tripartite graph and $\chi(G) \leq 3$. So we may assume that G contains no $K_{3,3}$.

Suppose that G has a proper 2-cutset $\{a, b\}$. So $G \setminus \{a, b\}$ can be partitioned into two sets X, Y such that there is no edge between them. Since G has no clique cutset, it is 2-connected, so there exists a path P_Y with ends a and b and with interior in Y. Let G'_X be the subgraph of G induced by $X \cup P_Y$. Note that P_Y is a flat path in G'_X. Let G''_X be obtained from G'_X by reducing P_Y. Define a graph G''_Y similarly. Since G'_X is an induced subgraph of G, it contains no ISK4. So, by Lemma 2.1.11, G''_X contains no ISK4. The same holds for G''_Y. By induction hypothesis, G''_X and G''_Y admit a 24-coloring. Since a and b have different colors in both coloring, we can combine them so that they coincide on $\{a, b\}$ and obtain a 24-coloring of G. Now, we may assume that G has no proper 2-cutset. If G contains a $K_{2,2,2}$ (rich square) or a prism, then by Lemma 2.1.10, G is the line graph of a graph with maximum degree 3, or a rich square, and by Lemmas 2.1.25 and 2.1.26, $\chi(G) \leq 4 < 24$. Therefore, we may assume that G contains neither prism nor $K_{2,2,2}$. So $G \in \mathcal{C}_3$ and $\chi(G) \leq 24$ by Lemma 2.1.24. □

2.1.5 Conclusion

The bound found in Theorem 2.1.7 is not only close to the one stated in Conjecture 2.1.2, but the simple structure of each layer is also interesting. We believe that it is very promising to settle Conjecture 2.1.2 by this way of looking at our class. For Theorem 2.1.8, we are convinced that the bound 24 we found could be slightly improved by this method if we look at each layer more carefully and exclude more structures, but it seems hard to reach the bound mentioned in Conjecture 2.1.1.

In the course of reviewing this result, Conjecture 2.1.2 has been proved in [12]. Moreover, it was done by settling also Conjecture 2.1.6. However, we prefer to keep our proof for Theorem 2.1.7 because of its nice and simple structure.

2.2 Detecting an ISK4

2.2.1 Introduction

We extend the notion of ISK4 to ISKn. For $n \geq 3$, an *ISKn* is a subdivision of a K_n. A graph that does not contain any subdivision of K_n as an induced subgraph is *ISKn-free*. A *twin wheel* is a graph consisting of a hole C and a vertex with exactly three consecutive neighbors in C. Note that K_4 and twin wheels are two special kinds of ISK4.

For a fixed graph H, one can check in polynomial time whether some graph G contains a subdivision of H as a *subgraph*, as a consequence of the results of the Graph Minors series [43]. However, checking whether G contains a subdivision of H as an *induced subgraph* is much more complicated. The question of detecting an induced subdivision of H (when H is fixed) in a given graph has been studied in [37]. There are certain graphs H where the problem is known to be NP-hard and graphs H where there exists a polynomial-time algorithm. For example, detecting an induced subdivision of K_3 is trivial since a graph is ISK3-free iff it is a forest. On the other hand, detecting an induced subdivision of K_5 has been shown to be NP-hard [37]. So far, apart from the trivial cases, we only know a polynomial-time algorithm to detect an induced subdivision of two subcubic graphs, which are $K_{2,3}$ [14] (subsivision of $K_{2,3}$ is also known as theta) and net [15]. In this section, we answer the question of detecting an ISK4, which was asked in [37] and [15]. For convenience in complexity analysis, we always denote by n the number of vertices of the input graph.

Theorem 2.2.1. *There is an algorithm with the following specifications:*

- **Input**: *Graph G.*
- **Output**:
  - *an ISK4 in G, or*
  - *the conclusion that G is ISK4-free.*
- **Running time**: $O(n^9)$.

Note that we have this theorem and several lemmas which correspond to some algorithms. Therefore, it depends on the context that we may call them theorems (lemmas) or algorithms.

2.2.2 Preliminaries

Let x, y, z be three distinct pairwise non-adjacent vertices in G. A graph H is an (x, y, z)-*radar* in G if it is an induced subgraph of G and:

- $V(H) = V(C) \cup V(P_x) \cup V(P_y) \cup V(P_z)$.
- C is an induced cycle of length ≥ 3 containing three distinct vertices x', y', z'.
- P_x is a path from x to x', P_y is a path from y to y', P_z is a path from z to z'.
- P_x, P_y, P_z are vertex-disjoint and x', y', z' are the only common vertices between them and C.
- These are the only edges in H.

Note that the length of each path P_x, P_y, P_z could be 0, therefore an induced cycle in G passing through x, y, z is also considered as an (x, y, z)-radar. Note also that an (x, y, z)-radar where the cycle C is a triangle, is a confluence of $\{x, y, z\}$ of Type 2. We now prove that Theorem 2.2.1 is a direct consequence of the following lemma.

Lemma 2.2.2. *There is an algorithm with the following specifications:*

- **Input**: *A graph G, four vertices $u, x, y, z \in V(G)$ such that $\{u, x, y, z\}$ induces a claw with center u in G.*
- **Output**: *One of the followings:*
  - *An ISK4 in G, or*

  - *Conclude that there is no (x, y, z)-radar in $G' = G \setminus (N[u] \setminus \{x, y, z\})$.*

- **Running time***: $O(n^5)$.*

Lemma 2.2.3. *An ISK4 is either K_4, a twin wheel or contains a claw.*

Proof of Theorem 2.2.1 by Lemma 2.2.2. We describe an algorithm to detect an ISK4 in G as follows. First, we check if there is a K_4 or a twin wheel in G. Checking if there exists a K_4 takes $O(n^4)$. Checking if there is a twin wheel in G can be done as follows: list all 4-tuples (a, b, c, d) of vertices in G such that they induce a $K_4 \setminus e$ (a graph obtained from K_4 by removing one edge, usually called a *diamond*) where $ad \notin E(G)$; for each tuple, check if a and d are connected in $G \setminus ((N[b] \cup N[c]) \setminus \{a, d\})$. Since we have $O(n^4)$ such tuples, this can be done in $O(n^6)$. If there exists a K_4 or a twin wheel in G, then output that ISK4 in G. Otherwise, move on to next step.

Now we may assume that G is $\{K_4$, twin wheel$\}$-free. The following claim is true thanks to Lemma 2.2.3: G contains an ISK4 iff there exists some 4-tuple (u, x, y, z) of vertices in G such that they induce a claw with center u and there is an (x, y, z)-radar in G'. The last step in our algorithm is the following: generate every 4-tuple (u, x, y, z) of vertices in G such that they induce a claw with center u and run Algorithm 2.2.2 for each tuple. If for some tuple (u, x, y, z), we detect an ISK4 in G then output that ISK4 and stop. If for all the tuples, we conclude that there is no (x, y, z)-radar in G' then we can conclude that G contains no ISK4. Since we have $O(n^4)$ such tuples, and it takes $O(n^5)$ for each tuple by Algorithm 2.2.2, the running time of our algorithm is $O(n^9)$. □

The rest of this chapter is therefore devoted to the proof of Lemma 2.2.2. In the next section, we introduce some useful structures and the main proof is presented in Section 2.2.4.

2.2.3 Antennas and cables

First we introduce two useful structures in our algorithm.

Let x, y, z be three distinct pairwise non-adjacent vertices in G. An (x, y, z)-*antenna* in G is an induced subgraph H of G such that:

- $V(H) = \{c\} \cup V(P_x) \cup V(P_y) \cup V(P_z)$.
- $c \notin \{x, y, z\} \cup V(P_x) \cup V(P_y) \cup V(P_z)$.
- P_x is a path from x to x', P_y is a path from y to y', P_z is a path from z to z'.

- P_x, P_y, P_z are vertex-disjoint and at least one of them has length ≥ 1.
- $cx', cy', cz' \in E(H)$.
- These are the only edges in H.
- For any vertex v in $G \setminus H$:
  - v has no neighbor in H or exactly one neighbor in H, or
  - v has exactly two neighbors v_1, v_2 in H such that for some $t \in \{x, y, z\}$, $v_1, v_2 \in P_t \cup \{c\}$ and their distance in H is 1 (so they are adjacent) or 2.

We also define *cable* given three distinct pairwise non-adjacent vertices x, y, z in G. An (x, y, z)-*cable* in G is an induced subgraph H of G such that:

- H is a path from x' to z' going through y' for some permutation (x', y', z') of $\{x, y, z\}$.
- For any vertex v in $G \setminus H$:
  - v has no neighbor in H or exactly one neighbor in H, or
  - v has exactly two neighbors v_1, v_2 in H such that for some $t \in \{x', z'\}$, v_1, v_2 are in the path $y'Ht$ and their distance in H is 1 or 2, or
  - v has exactly three neighbors in H, which are y' and two neighbors of y' in H.

Note that the existence of an (x, y, z)-antenna or (x, y, z)-cable in G implies that there is no vertex in G adjacent to all three vertices x, y, z.

The Steiner problem is the following. Given a graph $G = (V, E)$ with non-negative edge weights and let $S \subseteq V$ be a subset of the vertices, called *terminals*. The task is to find a minimum-weight Steiner tree - a tree in G that spans S. This problem is well-known to be NP-hard in general [33]. We will use the following algorithm, which is a direct consequence of Steiner problem in graphs for a fixed number of terminals:

Lemma 2.2.4. *There is an algorithm with the following specifications:*

- **Input***: A graph G, a subset $X \subseteq V(G)$ of size k (k is fixed).*
- **Output***: A minimum subgraph of G connecting every vertex in X (minimum with respect to the number of vertices).*
- **Running time***: $O(n^3)$.*

Proof. By considering X as the set of terminals and the weight of every edge is 1, the solution for Steiner problem in G with k terminals gives a tree T (a subgraph of G) connecting X with minimum number of edges. Since T is a tree, the number of its vertices differs exactly one from the number of its edges, therefore graph G induced by $V(T)$ is also a solution for the problem in Lemma 2.2.4. An $O(n^3)$ algorithm for Steiner problem in graphs with fixed number of terminals is given in [24]. □

We refer the reader to Section 2.1.2 for the definition of a confluence.

Lemma 2.2.5. *Given a connected graph G and three vertices $x, y, z \in V(G)$, a minimum subgraph H of G connecting x, y, z induces either:*

1. *A confluence of $\{x, y, z\}$ of Type 1, or*
2. *A confluence of $\{x, y, z\}$ of Type 2.*

Proof. If there are more edges, we would find a smaller subgraph in G connecting x, y, z, a contradiction. □

From now on, we always denote by G, u, x, y, z the input of Algorithm 2.2.2 and denote by G' the graph $G \setminus (N[u] \setminus \{x, y, z\})$. The following algorithm shows that we can detect some nice structures in G' in polynomial time.

Lemma 2.2.6. *There is an algorithm with the following specifications:*

- **Input***: G, u, x, y, z.*
- **Output***: One of the followings:*
  - *An ISK4 in G, or*
  - *Conclude that there is no (x, y, z)-radar in G', or*
  - *A vertex $v \in G'$ adjacent to all three vertices x, y, z, or*
  - *An (x, y, z)-antenna H in G', or*
  - *An (x, y, z)-cable H in G'.*
- **Running time***: $O(n^3)$.*

Proof. First, we check if x, y, z are connected in G' in $O(n^2)$. If they are not connected, conclude that there is no (x, y, z)-radar in G'. Now suppose that they are connected, we can find a minimum induced subgraph H of G connecting x, y, z by Algorithm 2.2.4. By Lemma 2.2.5, if H is a confluence of Type 2, output $H \cup \{u\}$

as an ISK4 in G. Therefore, we may assume that H is a confluence of Type 1. If H contains a vertex adjacent to both x, y, z, output that vertex and stop. Otherwise, we will prove that H must be an (x, y, z)-antenna or an (x, y, z)-cable in G', or G contains an ISK4. It is clear that now H must have the same induced structure as an antenna or a cable. We are left to prove that the attachment of a vertex $v \in G' \setminus H$ also satisfies the conditions in both cases:

- Case 1: H has the same induced structure as an (x, y, z)-antenna. Let c be the center of the only claw in H. Let x', y', z' be three neighbors of c such that x' (y', z') is the one closest to x (y, z, respectively) in H. Denote by P_x, P_y, P_z the paths from x to x', y to y', z to z' in H, respectively. Let $v \in G' \setminus H$. The following is true:

  - v cannot have neighbors in both P_x, P_y, P_z.

    If v does, $N_H(v) = \{x', y', z', c\}$ or $N_H(v) = \{x', y', z'\}$, otherwise $(H \setminus \{c, t\}) \cup \{v\}$ is a graph connecting x, y, z which is smaller than H, where t is one of $\{x', y', z'\}$, a contradiction. If $N_H(v) = \{x', y', z', c\}$, $\{u, v, c\} \cup P_x \cup P_y$ induces an ISK4 in G. If $N_H(v) = \{x', y', z'\}$, suppose that $z' \neq z$ (since v is not adjacent to both x, y, z), then $\{u, v, c, z'\} \cup P_x \cup P_y$ induces an ISK4 in G.
  - v has at most two neighbors in $P_x \cup \{c\}$ (this holds for P_y, P_z also).

    If v has at least four neighbors in $P_x \cup \{c\}$, let P be a shortest path from x to c in $H \cup \{v\}$, then $P \cup P_y \cup P_z$ induces a graph connecting x, y, z which is smaller than H, a contradiction. If v has exactly three neighbors in $P_x \cup \{c\}$, suppose that v has no neighbor in P_z (since v cannot have neighbors in both P_x, P_y, P_z), then $\{u, v, c\} \cup P_x \cup P_z$ induces an ISK4 in G.
  - v cannot have neighbors in both two paths among P_x, P_y, P_z.

    W.l.o.g, suppose v has neighbors in both P_x and P_y, we might assume that v has no neighbor in P_z. If v has two neighbors in one of $P_x \cup \{c\}$ and $P_y \cup \{c\}$, suppose that is $P_x \cup \{c\}$, let t be the neighbor of v in P_y which is closest to c. In this case, $\{u, v\} \cup P_x \cup P_z \cup tP_y y'$ induces an ISK4 in G. Therefore, v has exactly one neighbor in P_x and one neighbor in P_y and $H \cup \{u, v\}$ induces an ISK4 in G.
  - If v has exactly two neighbors in $P_x \cup \{c\}$, they must be of distance 1 or 2 in H.

    Otherwise, we find a graph connecting x, y, z smaller than H, a contradiction.

- Case 2: H has the same induced structure as an (x, y, z)-cable. Suppose that H is a path from x to z going through y. Let x', z' be the two neighbors of y in H

such that x' is closer to x in H. Denote by P_x, P_z the paths from x to x', z to z' in H, respectively. Let $v \in G' \setminus H$. The following is true:

- v has at most two neighbors in $P_x \cup \{y\}$.
  If v has four neighbors in $P_x \cup \{y\}$, let P be a shortest path from x to y in $H \cup \{v\}$, then $P \cup P_z$ is a subgraph connecting x, y, z which is smaller than H, a contradiction. If v has exactly three neighbors in $P_x \cup \{y\}$, then $\{u, v, y\} \cup P_x$ induces an ISK4 in G.
- If v has neighbors in both P_x, P_z, then $N_H(v) = \{x', y, z'\}$.
  We first show that v is adjacent to y. Suppose that v is not adjacent to y. If v has two neighbors in P_x, let t be the neighbor of v in P_z which is closest to y. In this case, $\{u, v, y\} \cup P_x \cup tP_z z'$ induces an ISK4 in G. Therefore, v has exactly one neighbor in P_x and one neighbor in P_y. But now, $\{u, v\} \cup H$ induces an ISK4 in G.
  Now, v is adjacent to y. Since v has at most two neighbors in $P_x \cup \{y\}$ and two neighbors in $P_z \cup \{y\}$, v has exactly one neighbor in P_x and one neighbor in P_z. If v is not adjacent to x', let t be the neighbor of v in P_x. Now $\{u, v, y\} \cup P_z \cup xP_x t$ induces an ISK4 in G. Therefore, $N_H(v) = \{x', y, z'\}$.
- If v has exactly two neighbors in P_x, they must be of distance 1 or 2 in H.
  Otherwise, we find a graph connecting x, y, z smaller than H, a contradiction.

□

Actually, there is an alternative way to implement Algorithm 2.2.6 more efficiently by not using Algorithm 2.2.4. Basically, we only have to consider a shortest path P_{xy} from x to y, then find a shortest path from z to P_{xy}. By that we would obtain immediately an (x, y, z)-antenna or (x, y, z)-cable. However, we use Algorithm 2.2.4 since it gives us a more convenient proof. The first case we need to handle in Algorithm 2.2.6 is when there is some vertex v adjacent to both x, y and z.

Lemma 2.2.7. *There is an algorithm with the following specifications:*

- **Input***: G, u, x, y, z, some vertex $v \in G'$ adjacent to x, y, z.*
- **Output***: One of the followings:*
  - *An ISK4 in G, or*
  - *Conclude that v is not contained in any (x, y, z)-radar in G'.*
- **Running time***: $O(n^2)$.*

Proof. It is not hard to see the following: v is contained in some (x, y, z)-radar in G' iff there exists a path from y to z in $G_x = G' \setminus ((N[x] \cup N[v]) \setminus \{y, z\})$ (up to a relabeling of x, y, z). Therefore, we only have to test if y and z are connected in G_x (and symmetries). If we find some path P from y to z in G_x, output $\{u, x, v\} \cup P$ as an ISK4. If no such path exists, we can conclude that v is not contained in any (x, y, z)-radar in G'. Since we only have to test the connection three times (between y and z in G_x, and symmetries), the running time of this algorithm is $O(n^2)$. □

We also have the following algorithm to handle with antenna.

Lemma 2.2.8. *There is an algorithm with the following specifications:*

- **Input***: G, u, x, y, z, an (x, y, z)-antenna H in G'.*
- **Output***: One of the followings:*
  - *An ISK4 in G, or*
  - *Conclude that there is no (x, y, z)-radar in G', or*
  - *Some vertex $c \in G'$ which is not contained in any (x, y, z)-radar in G'.*
- **Running time***: $O(n^4)$.*

Proof. Denote by c, x', y', z', P_x, P_y, P_z the elements of H as in the definition of an antenna. In this proof, we always denote by $N(X)$ the neighbor of X in G'. First, we prove that any path connecting any pair of $\{x, y, z\}$ in $G' \setminus c$ which contains at most two neighbors of c certifies the existence of an ISK4 in G. Such a path can be found by generating every pair (v, t) of neighbors of c in G', and for each pair, find a shortest path between each pair of $\{x, y, z\}$ in $G' \setminus (N[c] \setminus \{v, t\})$. It is clear that if such a path is found by this algorithm, then it has at most two neighbors of c and if no path is reported, we can conclude that it does not exist. Since we have $O(n^2)$ pairs (v, t) and finding a shortest path between some pair of vertices in a graph takes $O(n^2)$, this algorithm runs in $O(n^4)$. Now we prove that such a path certifies the existence of an ISK4. Let P be a path between some pair in $\{x, y, z\}$ that contains at most two neighbors of c, w.l.o.g assume that P is from x to y. We say that a path Q is a (P_x, P_y)-*connection* if one end of Q is in $N(P_x)$, the other end is in $N(P_y)$ and $Q^* \cap (N(P_x) \cup N(P_y)) = \emptyset$ (we make symmetric definitions for (x, z) and (y, z)). We also say that a path Q is S-*independent* for some $S \subseteq V(G')$ if $Q \cap N[S] = \emptyset$. We consider following cases:

1. P contains no neighbor of c.

   It is clear that there exists a subpath P' of P such that P' is a (P_x, P_y)-connection. Furthermore, we may assume that P' is P_z-independent since otherwise there exists some subpath P'' of P' which is a (P_x, P_z)-connection and is P_y-independent. Let x'', y'' be two ends of P' which are in $N(P_x)$ and $N(P_y)$, respectively. In this case x'' and y'' are not adjacent to c since P' contains no neighbor of c. We have the following cases based on the attachment on an antenna:

   (a) x'' and y'', each has exactly one neighbor in P_x and P_y, respectively. Then $\{u\} \cup P' \cup H$ induces an ISK4 in G.

   (b) x'' has exactly one neighbor in P_x and y'' has exactly two neighbors in P_y (or symmetric). Then $\{u, c\} \cup P' \cup P_x \cup P_y$ induces an ISK4 in G.

   (c) x'' and y'', each has exactly two neighbors in P_x and P_y, respectively. Let t be the neighbor of x in P_x which is closer to c. Then $\{u, c\} \cup P' \cup tP_xx' \cup P_y \cup P_z$ induces an ISK4 in G.

2. P contains exactly one neighbor of c.

   Similar to the argument of the previous case, there exists a path P' with two ends x'' and y'' such that P' is a (P_x, P_y)-connection and is P_z-independent. We may assume that c has exactly one neighbor c' in P', otherwise we are back to previous case. In this case, at most one vertex in $\{x'', y''\}$ can be adjacent to c (in other words, at most one vertex in $\{x'', y''\}$ can be identical to c'). We consider the following cases:

   (a) Each of $\{x'', y''\}$ has exactly one neighbor in $P_x \cup \{c\}$, and therefore exactly one neighbor in P_x. Then $\{u, c\} \cup P' \cup P_x \cup P_y$ induces an ISK4 in G.

   (b) x'' has exactly one neighbor in $P_x \cup \{c\}$ (this neighbor must be in P_x) and y'' has exactly two neighbors in $P_x \cup \{c\}$ (or symmetric). If y'' is adjacent to c, then $\{u, c\} \cup P' \cup P_x \cup P_y$ induces an ISK4 in G. Otherwise y'' has two neighbors in P_y and $\{u, c\} \cup c'P'y'' \cup P_x \cup P_y$ induces an ISK4 in G.

   (c) Each of $\{x'', y''\}$ has exactly two neighbors in $P_x \cup \{c\}$. Since at most one of them is adjacent to c, we might assume that y is not adjacent to c. Then $\{u, c\} \cup c'P'y'' \cup P_y \cup P_z$ induces an ISK4 in G.

3. P contains exactly two neighbors of c.

   We may assume that P is P_z-independent since otherwise we have some subpath of P which is a (P_z, P_x)-connection or (P_z, P_y)-connection and contains at most

one neighbor of c that we can argue like previous cases. Therefore, $\{u, c\} \cup P \cup P_z$ induces an ISK4.

It is easy to see that above argument can be turned into an algorithm to output an ISK4 in each case. Now we can describe our algorithm for Lemma 2.2.8. First, test if there exists a path in $G' \setminus c$ between some pair of $\{x, y, z\}$ which contains at most two neighbors of c:

1. If such a path exists, output the corresponding ISK4 in G.
2. If no such path exists, test the connection between each pair of $\{x, y, z\}$ in $G' \setminus c$:
   (a) If $\{c\}$ is a cutset in G' disconnecting some pair of $\{x, y, z\}$, then conclude that there is no (x, y, z)-radar in G'.
   (b) Otherwise, conclude that c is the vertex not contained in any (x, y, z)-radar in G'.

Now we explain why this algorithm is correct. If Case 1 happens, it outputs correctly an ISK4 by the argument above. If Case 2 happens, we know that there are only two possible cases for the connection between each pair of $\{x, y, z\}$ in $G' \setminus c$, for example for (x, y):

- x and y are not connected in $G' \setminus c$, or
- Every path from x to y in $G' \setminus c$ contains at least three neighbors of c.

Therefore, Case 2a corresponds to one of the following cases, both lead to the conclusion that there is no (x, y, z)-radar in G':

- Each pair of $\{x, y, z\}$ is not connected in $G' \setminus c$.
- x is not connected to $\{y, z\}$, while y and z are still connected in $G' \setminus c$ (or symmetric). In this case every path from y to z in $G' \setminus c$ contains at least three neighbors of c.

If Case 2b happens, we know that each pair of $\{x, y, z\}$ is still connected in $G' \setminus c$ and furthermore every path between them contains at least three neighbors of c. This implies that c is not contained in any (x, y, z)-radar, since if c is in some (x, y, z)-radar, we can easily find a path between some pair of $\{x, y, z\}$ in that radar containing at most two neighbors of c, a contradiction.

The complexity of the whole algorithm is still $O(n^4)$ since we can find an ISK4 in Case 1 in $O(n^2)$ and test the connection in Case 2 in $O(n^2)$. □

The next algorithm deals with cable.

Lemma 2.2.9. *There is an algorithm with the following specifications:*

- **Input***: G, u, x, y, z, an (x, y, z)-cable H in G'.*
- **Output***: One of the followings:*
  - *An ISK4 in G, or*
  - *Conclude that there is no (x, y, z)-radar in G', or*
  - *Some vertex $c \in G'$ which is not contained in any (x, y, z)-radar in G'.*
- **Running time***: $O(n^4)$.*

Proof. W.l.o.g we can assume that cable H is a path from x to z containing y. Let x' be the neighbor of y in H which is closer to x and z' be the other neighbor of y in H. Let $P_x = xHx'$ and $P_z = zHz'$. In this proof, we denote by $N(X)$ the neighbor of X in G'. We also say that a path Q is a (P_x, P_z)-*connection* if one end of Q is in $N(P_x)$, the other end is in $N(P_z)$ and $Q^* \cap (N(P_x) \cup N(P_z)) = \emptyset$. Before the algorithm, we first prove the followings:

(1) Every path P from x to z in $G' \setminus y$ containing no neighbor of y certifies an ISK4 in G.

Let P' be a subpath of P such that P' is a (P_x, P_z)-connection. Let x'' and z'' be two ends of P' such that $x'' \in N(P_x)$ and $z'' \in N(P_z)$. Since P' has no neighbor of y, both x'' and z'' are not adjacent to y. We consider the following cases based on the attachment on a cable:

(a) x'' and z'', each has exactly one neighbor in P_x and P_z, respectively. Then $\{u\} \cup H \cup P'$ induces an ISK4 in G.

(b) x'' has exactly two neighbors in P_x and z'' has exactly one neighbor in P_z (or symmetric). Let t be the neighbor of z'' in P_z.

- If $t \neq z$ then $\{u, y\} \cup P' \cup P_x \cup tP_zz'$ induces an ISK4 in G.
- If $t = z$ then $\{u, y, z\} \cup P' \cup P_x$ induces an ISK4 in G.

(c) x'' and z'', each has exactly two neighbors in P_x and P_z, respectively. Let t be one of the two neighbors of z'' which is closer to y. Then $\{u, y\} \cup P' \cup P_x \cup tP_zz'$ induces an ISK4 in G.

(2) Every path P from x to z in $G' \setminus y$ containing exactly two neighbors of y certifies an ISK4 in G.

It is clear since $\{u, y\} \cup P$ induces an ISK4 in G.

(3) Assume that every path from x to z in $G' \setminus y$ contains at least one neighbor of y. If there exists some path from x to z in $G' \setminus y$ containing exactly one neighbor of y, then a shortest such path P satisfies that $P \cup \{y\}$ is an (x, y, z)-antenna in G', or G contains an ISK4.

It is clear that $P \cup \{y\}$ has the same induced structure as an antenna, we only have to prove the attachment on it. Let c be the only neighbor of y on P and x', z' be the two neighbors of c different from y such that x' is the one closer to x in P. Denote $P_x = xPx'$, $P_z = zPz'$. Let v be a vertex in $G' \setminus (P \cup \{y\})$, we consider the following cases:

- v is not adjacent to y. The following is true:
  - v cannot have neighbors on both P_x and P_z.
    If v does, there exists a path in G' from x to z (passing through v) containing no neighbor of y, a contradiction.
  - v has at most two neighbors in $P_x \cup \{c\}$.
    If v has at least three neighbors in $P_x \cup \{c\}$, they must be exactly three consecutive neighbors in P, otherwise there exists a shorter path than P satisfying the assumption. But if v has three consecutive neighbors in $P_x \cup \{c\}$, then $\{u, v\} \cup P$ induces an ISK4.
  - If v has exactly two neighbors in $P_x \cup \{c\}$, they must be of distance 1 or 2 in P.
    Otherwise, we have a shorter path than P (passing through v) satisfying the assumption.
- v is adjacent to y. The following is true:
  - v cannot have neighbors on both P_x and P_z.
    If v does, $N(v) \cap P_x = \{x'\}$ and $N(v) \cap P_z = \{z'\}$, otherwise there exists a shorter path than P (passing through v) satisfying the assumption. If v is adjacent to c, then $\{u, v\} \cup P$ induces an ISK4. If v is not adjacent to c, since v cannot adjacent to both x and z (by definition of cable), assume v is not adjacent to x (or equivalently $x \neq x'$). In this case, $\{u, v, y, x'\} \cup P_z$ induces an ISK4 in G.
  - v cannot have at least three neighbors in $P_x \cup \{c\}$.

If v does, there exists a path (passing through v) from x to z containing exactly two neighbors of y (which are v and c). This path certifies an ISK4 by (2).

- v cannot have exactly two neighbors in $P_x \cup \{c\}$.
  If v does, $\{u, v, y, c\} \cup P_x$ induces an ISK4.
- If v has exactly one neighbor in $P_x \cup \{c\}$, it must be c.
  If v has exactly one neighbor in $P_x \cup \{c\}$ which is not c, then $\{u, v, y\} \cup P$ induces an ISK4.

The above discussion shows that either G contains an ISK4 (and we can detect in $O(n^2)$), or $P \cup \{y\}$ is an (x, y, z)-antenna in G'.

Now we describe our algorithm for Lemma 2.2.9:

1. Test if there exists a path P from x to z in $G' \setminus y$ containing no neighbor of y.
   (a) If such a path exists, output an ISK4 by the argument in (1).
   (b) If no such path exists, move to the next step.
2. Find a shortest path P from x to z in $G' \setminus y$ containing exactly one neighbor of y if such a path exists.
   (a) If such a path P exists, by the argument in (3), either we detect an ISK4 in G, output it and stop, or we find an (x, y, z)-antenna $P \cup \{y\}$ in G', run Algorithm 2.2.8 with this antenna as input, output the corresponding conclusion.
   (b) If no such path exists, move to the next step.
3. Test if there exists a path P from x to z in $G' \setminus y$ containing exactly two neighbors of y.
   (a) If such a path P exists, output an ISK4 in G by the argument in (2).
   (b) If no such path exists, conclude there is no (x, y, z)-radar in G' (Since at this point, every path from x to z in $G' \setminus y$ contains at least three neighbors of y).

Step 1 can be done in $O(n^2)$ by checking the connection between x and z in $G' \setminus N[y]$. Step 2 runs in $O(n^3)$ by generating every neighbor t of y and for each t, find a shortest path between x and z in $G' \setminus (N[y] \setminus \{t\})$. And pick the shortest one over all such paths. Since we call the Algorithm 2.2.8, step 2a takes $O(n^4)$. Step 3

can be done in $O(n^4)$ by generating every pair (t, w) of neighbors of y and for each pair (t, w), check the connection between x and z in $G' \setminus (N[y] \setminus \{t, w\})$. Therefore, the total running time of Algorithm 2.2.9 is $O(n^4)$. □

2.2.4 Proof of Lemma 2.2.2

Now we sum up everything in previous section and describe the algorithm for Lemma 2.2.2:

1. Run Algorithm 2.2.6. Output is one of the followings:
   (a) An ISK4 in G: output it and stop.
   (b) Conclude that there is no (x, y, z)-radar in G' and stop.
   (c) A vertex v adjacent to x, y, z: run Algorithm 2.2.7 with v as input. Output is one of the followings:
      i. An ISK4 in G: output it and stop.
      ii. Conclude that v is not contained in any (x, y, z)-radar in G': Run Algorithm 2.2.2 recursively for $(G \setminus v, u, x, y, z)$.
   (d) An (x, y, z)-antenna H in G': run Algorithm 2.2.8 with H as input. Output is one of the followings:
      i. An ISK4 in G: output it and stop.
      ii. Conclude that there is no (x, y, z)-radar in G' and stop.
      iii. Some vertex $c \in G'$ which is not contained in any (x, y, z)-radar in G': Run Algorithm 2.2.2 recursively for $(G \setminus c, u, x, y, z)$.
   (e) An (x, y, z)-cable H in G': run Algorithm 2.2.9 with H as input. Output is one of the followings:
      i. An ISK4 in G: output it and stop.
      ii. Conclude that there is no (x, y, z)-radar in G' and stop.
      iii. Some vertex $c \in G'$ which is not contained in any (x, y, z)-radar in G': Run Algorithm 2.2.2 recursively for $(G \setminus c, u, x, y, z)$.

The correctness of this algorithm is based on the correctness of the Algorithms 2.2.6, 2.2.7, 2.2.8 and 2.2.9. Now we analyse its complexity. Let $f(n)$ be the complexity of this algorithm. Since we have five cases, each case takes $O(n^4)$ and at most a recursive call with the complexity $f(n-1)$. Therefore $f(n) \leq O(n^4) + f(n-1)$ and $f(n) = O(n^5)$.

2.2.5 Conclusion

In this section, we give an $O(n^9)$ algorithm to detect an induced subdivision of K_4 in a given graph. We believe that the complexity might be improved to $O(n^7)$ by first decomposing the graph by clique cutset until there is no $K_{3,3}$ (using decomposition theorem in [38]). Now every (ISK4, $K_{3,3}$)-free graph has a linear number of edges since it is c-degenerate by some constant c as shown in [38]. Therefore, testing the connection in this graph takes only $O(n)$, instead of $O(n^2)$ as in the algorithm. Also, we only have to consider $O(n^3)$ triples of three independent vertices and test every possible center of that claw at the same time instead of generating all $O(n^4)$ claws. But we prefer to keep our algorithm as $O(n^9)$ since it is simple and does not rely on decomposition theorem. We leave the following open question as the conclusion:
Open question. Given a graph H of maximum degree 3, can we detect an induced subdivision of H in polynomial time?

Chapter 3

Even-hole-free graphs

A graph is *even-hole-free* if it does not contain any hole of even length as an induced subgraph. The study of even-hole-free graphs was motivated by perfect graphs, since these two classes have a very close relation. In fact, the decomposition technique which was developed during the study of even-hole-free graphs led to the proof of Strong Perfect Graph Conjecture 1.2.1. In this chapter, we study even-hole-free graphs with no star cutset. We give the optimal upper bound for its chromatic number in terms of clique number in Section 3.3 and a polynomial-time algorithm to color any graph in this class in Section 3.4. The latter is, in fact, a direct consequence of our proof that this class has bounded rank-width. The results of this chapter are covered in the following paper:

[III] N.K. Le. Coloring even-hole-free graphs with no star cutset, *arXiv preprint*, submitted.

3.1 Introduction

The structure of even-hole-free graphs was first studied by Conforti, Cornuéjols, Kapoor and Vušković in [17] and [18]. They were focused on showing that even-hole-free graphs can be recognized in polynomial time, and their primary motivation was to develop techniques which can then be used in the study of perfect graphs. In [17], they obtained a decomposition theorem for even-hole-free graphs that uses 2-joins and star, double star and triple star cutsets, and in [18], they used it to obtain a polynomial time recognition algorithm for even-hole-free graphs. That decomposition technique is actually useful since the Strong Perfect Graph Conjecture was proved in [13] by decomposing Berge graphs using skew cutsets, 2-joins and their complements. Soon after, the recognition of Berge graphs was shown to be poly-

nomial by Chudnovsky, Cornuéjols, Liu, Seymour and Vušković in [11]. A better decomposition theorem for even-hole-free graphs using only 2-joins and star cutsets was given in [20].

Finding a maximum clique, a maximum independent set and an optimal coloring are all known to be polynomial for perfect graphs [28, 29]. However, these algorithms rely on the ellipsoid method, which is impractical. It is still an open question to find a combinatorial algorithm for these problems. On the other hand, the complexities of finding a maximum stable set and an optimal coloring are both open for even-hole-free graphs. Note that a maximum clique of an even-hole-free graphs can be found in polynomial time, since a graph without a hole of length 4 has polynomial number of maximal cliques and one can list them all in polynomial time [26].

Therefore, we would like to see if the decomposition theorem can be used to design polynomial-time algorithms for all these combinatorial problems. The general answer should be impossible since there are some kinds of decomposition which do not seem to be friendly with these problems like star or skew cutsets. On the other hand, 2-joins look very promising. Indeed, in [48], Trotignon and Vušković already gave the polynomial algorithms to find a maximum clique and maximum independent set in the subclasses of even-hole-free and Berge graphs which are fully decomposable by only 2-joins (namely, even-hole-free graphs with no star cutset and perfect graphs with no balanced skew-partition, homogenous pair nor complement 2-join). In [16], they generalize the result for Berge graphs to perfect graph with no balanced skew-partitions. Note that an $O(n^k)$ algorithm that computes a maximum weighted independent set for a class of perfect graphs closed under complementation, yields also an $O(n^{k+2})$ algorithm that computes an optimal coloring for the same class (see for instance [35, 44]). Hence, all three problems (clique, independent set and coloring) are solved for perfect graph with no balanced skew-partitions. However, the coloring problem for even-hole-free graphs with no star cutset remains open despite its nice structure. In this chapter, we prove that this class has bounded rank-width, a graph parameter which will be defined in the next section. This implies that it also has bounded clique-width (a parameter which is equivalent to rank-width in the sense that one is bounded if and only if the other is also bounded). Therefore, coloring is polynomial-time solvable for even-hole-free graphs with no star cutset by combining the two results: Kobler and Rotics [34] showed that for any constant q, coloring is polynomial-time solvable if a q-expression is given, and Oum [40] showed that a $(8^p - 1)$-expression for any n-vertex graph with clique-width at most p can be found in $O(n^3)$. Note that our result is strong in the sense that it implies that every graph problem expressible in monadic second-order logic formula is solvable in polynomial-time for even-hole-free graphs with no star cutset (including also finding

a maximum clique and a maximum independent set).

We also know that even-hole-free graphs are χ-bounded by the concept introduced by Gyárfás [30]. In [1], it is proved that $\chi(G) \leq 2\omega(G) - 1$ for every even-hole-free graph G. One might be interested in knowing whether this bound could be improved for the class that we are considering, even-hole-free graphs with no star cutset. Let $\text{rwd}(G)$ denote the rank-width of some graph G. The main results of this chapter are the two following theorems:

Theorem 3.1.1. *Let G be a connected even-hole-free graph with no star cutset. Then $\chi(G) \leq \omega(G) + 1$.*

Theorem 3.1.2. *Let G be a connected even-hole-free graph with no star cutset. Then $\text{rwd}(G) \leq 3$.*

The rest of this chapter is organized as follows. In Section 3.2, we formally define every notion and mention all the results that we use in this chapter. The proof of Theorem 3.1.1 is presented in Section 3.3 and the proof of Theorem 3.1.2 is given in Section 3.4.

3.2 Preliminaries

Our proof heavily relies on the decomposition lemmas for even-hole-free graphs with no star cutset given by Trotignon and Vušković in [48]. Hence, in the first part of this section, the formal definitions needed to state these lemmas will be given. We refer the reader to Section 1.3 of Chapter 1 for the definitions of several decompositions (star cutset, clique cutset, 2-join,...).

Since the goal of decomposition theorems is to break our graphs into smaller pieces that we can handle inductively, we need a way to construct them. *Blocks of decomposition* with respect to a 2-join (which will be defined below) are built by replacing each side of the 2-join by a path and the next lemma shows that for even-hole-free graphs, there exists a unique way to choose the parity of that path.

Lemma 3.2.1 ([48])**.** *Let G be an even-hole-free graph and $(X_1, X_2, A_1, B_1, A_2, B_2)$ be a split of a 2-join of G. Then for $i = 1, 2$, all the paths with an end in A_i, an end in B_i and interior in C_i have the same parity.*

Let G be an even-hole-free graph and $(X_1, X_2, A_1, B_1, A_2, B_2)$ be a split of a 2-join of G. The *blocks of decomposition* of G with respect to (X_1, X_2) are the two graphs G_1, G_2 built as follows. We obtain G_1 by replacing X_2 by a *marker path* P_2 of length k_2, from a vertex a_2 complete to A_1, to a vertex b_2 complete to B_1 (the interior of

P_2 has no neighbor in X_1). We choose $k_2 = 3$ if the length of all the paths with an end in A_2, an end in B_2 and interior in C_2 is odd (they have the same parity due to Lemma 3.2.1), and $k_2 = 4$ otherwise. The block G_2 is obtained similarly by replacing X_1 by a marker path P_1 of length k_1 with two ends a_1, b_1.

We refer the reader to Section 1.4 of Chapter 1 for the definition of a pyramid. Note that in an even-hole-free graph, the lengths of all three paths of a pyramid have the same parity. Now we introduce the last basic graph in our class.

An *extended nontrivial basic* graph R is defined as follows:

1. $V(R) = V(L) \cup \{x, y\}$.
2. L is the line graph of a tree T.
3. x and y are adjacent, $x, y \notin V(L)$.
4. Every maximal clique of size at least 3 in L is called an *extended* clique. L contains at least two extended cliques.
5. The nodes of L corresponding to the edges incident with vertices of degree one in T are called *leaf nodes*. Each leaf node of L is adjacent to exactly one of $\{x, y\}$, and no other node of L is adjacent to $\{x, y\}$.
6. These are the only edges in R.

Note that the definition of the extended nontrivial basic graph we give here is simplified compared to the one from the original paper [20] (since they prove a decomposition theorem for a more general class, namely, 4-hole-free odd-signable graphs), but it is all we need in our proof. The following property of R is easy to observe in even-hole-free graphs with no star cutset:

Lemma 3.2.2. *x (and y) has at most one neighbor in every extended clique. Furthermore, if x has some neighbor in an extended clique K, then $N(y) \cap K = \emptyset$.*

Proof. If x has two neighbors a, b in some extended clique K, then $N(a) \setminus \{b\} = N(b) \setminus \{a\}$, implying that there is a star cutset $S = (\{a\} \cup N(a)) \setminus \{b\}$ in R separating b from the rest of the graph, a contradiction. Also, if x and y both have a neighbor in a same extended clique, called a and b, respectively, then $\{x, a, b, y\}$ induces a 4-hole, a contradiction. □

An even-hole-free graph is *basic* if it is one of the following graphs:

- a clique,

- a hole,
- a long pyramid, or
- an extended nontrivial basic graph.

Now, we are ready to state the decomposition theorem for even-hole-free graphs given by Da Silva and Vušković.

Theorem 3.2.3 (Da Silva, Vušković [20]). *A connected even-hole-free graph is either basic or it has a 2-join or a star cutset.*

By this theorem, we already know that even-hole-free graphs with no star cutset always have a 2-join. But we might prefer something a bit stronger for our purpose. A 2-join is called *extreme* if one of its block of decomposition is basic. The two following lemmas (which can be found in Sections 3 and 4 in [48]) say that: our blocks of decomposition with respect to a 2-join remain in the class and our class is fully decomposable by extreme 2-joins. This is convenient for an inductive proof.

Lemma 3.2.4 (Trotignon, Vušković [48]). *Let G be a connected even-hole-free graph with no star cutset and (X_1, X_2) is a 2-join of G. Let G_1 be a block of decomposition with respect to this 2-join. Then G_1 is a connected even-hole-free graph with no star cutset.*

Lemma 3.2.5 (Trotignon, Vušković [48]). *A connected even-hole-free graph with no star cutset is either basic or it has an extreme 2-join.*

By Lemmas 3.2.4 and 3.2.5, we know that even-hole-free graphs with no star cutset can be fully decomposed into basic graphs using only extreme 2-joins. However, we need a little more condition to avoid confliction between these 2-joins, that is, every 2-join we use is *non-crossing*, meaning that every marker path in the process always lies entirely in one side of every following 2-joins (the edges between X_1 and X_2 do not belong to any marker path). Now we define the *2-join decomposition tree* for this purpose. Note that this definition we give here is not only for even-hole-free graphs with no star cutset, but also works in a more general sense. It is well defined for any graph class with its own basic graphs. Let $\mathbb{D}$ be a class of graphs and $\mathbb{B} \subseteq \mathbb{D}$ be the set of basic graphs in $\mathbb{D}$. Given a graph $G \in \mathbb{D}$, a tree $\mathbb{T}_G$ is a *2-join decomposition tree* for G if:

- Each node of $\mathbb{T}_G$ is a pair (H, S), where H is a graph in $\mathbb{D}$ and S is a set of disjoint flat paths of H.

- The root of $\mathbb{T}_G$ is $(G, \emptyset)$.
- Each non-leaf node of $\mathbb{T}_G$ is (G', S'), where G' has a 2-join (X_1, X_2) such that the edges between X_1 and X_2 do not belong to any flat path in S'. Let $S_1, S_2 \subseteq S'$ be the set of the flat paths of S' in $G'[X_1]$, $G'[X_2]$, respectively (note that $S' = S_1 \cup S_2$). Let G_1, G_2 be two blocks of decomposition of G' with respect to this 2-join with marker paths P_2, P_1, respectively. The node (G', S') has two children, which are $(G_1, S_1 \cup \{P_2\})$ and $(G_2, S_2 \cup \{P_1\})$.
- Each leaf node of $\mathbb{T}_G$ is (G', S'), where $G' \in \mathbb{B}$.

Note that by this definition, each set S' in some node (G', S') of $\mathbb{T}_G$ is properly defined in top-down order (from root to leaves). A 2-join decomposition tree is called *extreme* if each non-leaf node of it has a child which is a leaf node.

Lemma 3.2.6 (Trotignon, Vušković [48])**.** *Every connected even-hole-free graphs with no star cutset has an extreme 2-join decomposition tree.*

Observation 3.2.7. *Every block of decomposition with respect to a 2-join of a connected even-hole-free graph with no star cutset which is basic is either a long pyramid or an extended nontrivial basic graph.*

Let us review the definition of rank-width, which was first introduced in [41]. For a matrix $M = \{m_{ij} : i \in R, j \in C\}$ over a field F, let $\mathrm{rk}(M)$ denote its linear rank. If $X \subseteq R$, $Y \subseteq C$, then let $M[X, Y]$ be the submatrix $\{m_{ij} : i \in X, j \in Y\}$ of M. We assume that adjacency matrices of graphs are matrices over $GF(2)$.

Let G be a graph and A, B be disjoint subsets of $V(G)$. Let M be the adjacency matrix of G over $GF(2)$. We define the *rank* of (A, B), denoted by $\mathrm{rk}_G(A, B)$, as $\mathrm{rk}(M[A, B])$. The *cut-rank* of a subset $A \subseteq V(G)$, denoted by $\mathrm{cutrk}_G(A)$, is defined by

$$\mathrm{cutrk}_G(A) = \mathrm{rk}_G(A, V(G) \setminus A).$$

A *subcubic tree* is a tree such that the degree of every vertex is either one or three. We call (T, L) a *rank-decomposition* of G if T is a subcubic tree and L is a bijection from $V(G)$ to the set of leaves of T. For an edge e of T, the two connected components of $T \setminus e$ correspond to a partition $(A_e, V(G) \setminus A_e)$ of $V(G)$. The *width* of e of the rank-decomposition (T, L) is $\mathrm{cutrk}_G(A_e)$. The *width* of (T, L) is the maximum width over all edges of T. The *rank-width* of G, denoted by $\mathrm{rwd}(G)$, is the minimum width over all rank-decompositions of G (If $|V(G)| \leq 1$, we define $\mathrm{rwd}(G) = 0$).

Observation 3.2.8. *The rank-width of a clique is at most* 1 *and the rank-width of a hole is at most* 2*.*

3.3 χ-bounding function

3.3.1 Special graphs

Recall that the bound of chromatic number for even-hole-free graphs ($\chi(G) \leq 2\omega(G)-1$) is obtained by showing that there is a vertex whose neighborhood is a union of two cliques [1]. We would like to do the same things for our class. However, since our class is not closed under vertex-deletion, instead of showing that there exists a vertex whose neighborhood is "simple", we have to show that there is an elimination order such that the neighborhood of each vertex is "simple" in the remaining graph. To achieve that goal, we introduce *special* graphs. In fact, this is just a way of labeling vertices for the sake of an inductive proof.

A graph G is *special* if it is associated with a pair (C_G, F_G) such that:

- $C_G \subseteq V(G)$, $F_G \subseteq V(G)$ and $C_G \cap F_G = \emptyset$.
- Every vertex in F_G has degree 2.
- Every vertex in C_G has at least one neighbor in F_G.

Note that any graph can be seen as a special graph with $C_G = F_G = \emptyset$.

Suppose that G has some split $(X_1, X_2, A_1, B_1, A_2, B_2)$ of a 2-join. Due to this new notion of special graph, we want to specify the pairs (C_{G_1}, F_{G_1}) and (C_{G_2}, F_{G_2}) for the blocks of decomposition G_1, G_2 of G with respect to this 2-join to ensure that the two blocks we obtained are also special. Let $C_i = C_G \cap X_i$, $F_i = F_G \cap X_i$ $(i = 1, 2)$, we choose the pair (C_{G_1}, F_{G_1}) as follows:

- If $|A_1| = 1$, the only vertex in A_1 is in C_G and $A_2 \cap F_G \neq \emptyset$, then set $C_a = \emptyset$, $F_a = \{a_2\}$. Otherwise set $C_a = \{a_2\}$, $F_a = \emptyset$.
- If $|B_1| = 1$, the only vertex in B_1 is in C_G and $B_2 \cap F_G \neq \emptyset$, then set $C_b = \emptyset$, $F_b = \{b_2\}$. Otherwise set $C_b = \{b_2\}$, $F_b = \emptyset$.
- Finally, set $C_{G_1} = C_1 \cup C_a \cup C_b$, $F_{G_1} = F_1 \cup F_a \cup F_b \cup V(P_2^*)$.

The pair (C_{G_2}, F_{G_2}) for block G_2 is chosen similarly.

Lemma 3.3.1. *Let G be a special connected even-hole-free graph with no star cutset associated with (C_G, F_G) and $(X_1, X_2, A_1, B_1, A_2, B_2)$ be a split of a 2-join of G. Let G_1 be a block of decomposition with respect to this 2-join. Then G_1 is a special graph associated with (C_{G_1}, F_{G_1}).*

Proof. Remark that since G is 4-hole-free, one of A_1 and A_2 must be a clique (similar for B_1 and B_2). Now we prove that if one of A_1 and A_2 intersects F_G, then the other set is of size 1. Suppose that $A_1 \cap F_G \neq \emptyset$, we will prove that $|A_2| = 1$. Indeed, since $f \in A_1 \cap F_G$ has degree 2, $|A_2| \leq 2$. If $|A_2| = 2$ then f is the only vertex in A_1 (otherwise, A_2 must be a clique and $N(f)$ is a clique cutset separating f from the rest of G, a contradiction to the fact that G has no star cutset). Therefore, f has no neighbor in X_1, so there is no path from A_1 to B_1 in $G[X_1]$, a contradiction to the definition of a 2-join. This proves that $|A_2| = 1$. Now, G_1 is a special graph associated with (C_{G_1}, F_{G_1}) because:

1. Every vertex f in F_{G_1} has degree 2.

   If $f \in F_1 \setminus (A_1 \cup B_1)$, then degree of f remains the same in G and G_1. If $f \in F_1 \cap (A_1 \cup B_1)$, say $f \in F_1 \cap A_1$, from the above remark, $|A_2| = 1$, therefore the degree of f remains the same in G and G_1. If $f \in F_a \cup F_b$ then $|A_1| = 1$ by the way we choose F_{G_1}, so f has degree 2 in G_1. If $f \in P_2^*$, then it is an interior vertex of a flat path, therefore it has degree 2.

2. Every vertex c in C_{G_1} has at least a neighbor in F_{G_1}.

   If $c \in C_1$ and its neighbor in F_G is in X_1, then c has a neighbor in F_1. If $c \in C_1$ and its neighbor in F_G is in $A_2 \cup B_2$, say A_2, then its neighbor in F_{G_1} is a_2. If $c \in C_a \cup C_b$, then its neighbor in F_{G_1} is one of the two ends of P_2^*.

□

3.3.2 Elimination order

Let G be a special graph associated with (C_G, F_G). A vertex $v \in V(G)$ is *almost simplicial* if its neighborhood induces a clique or a union of a clique K and a vertex u such that $u \notin C_G$ (u can have neighbor in K). An elimination order $v_1, \ldots, v_k$ of vertices of $G \setminus F_G$ is *nice* if for every $1 \leq i \leq k$, v_i is *almost simplicial* in $G \setminus (F_G \cup \{v_1, \ldots, v_{i-1}\})$. The next lemma is the core of this section.

Lemma 3.3.2. *Let G be a special connected even-hole-free graph with no star cutset associated with (C_G, F_G). Then $G \setminus F_G$ admits a nice elimination order.*

By setting $C_G = F_G = \emptyset$, we have the following corollary of Lemma 3.3.2:

Corollary 3.3.3. *Let G be a connected even-hole-free graph with no star cutset. Then G admits a nice elimination order.*

Theorem 3.1.1 follows immediately from the above corollary since we can greedily color G in the reverse order of that nice elimination order using at most $\omega(G)+1$ colors. Therefore, the rest of this section is to devoted to the proof Lemma 3.3.2.

Lemma 3.3.4. *Let G be a special basic even-hole-free graph with no star cutset associated with (C_G, F_G) and G is neither a clique nor a hole. Let P be a flat path of length at least 2 in G. We denote by u_1, u_2 the two ends of P.*

- *If $N(u_1) \setminus V(P)$ is a clique, set $K_1 = N(u_1) \setminus V(P)$, otherwise set $K_1 = \emptyset$.*
- *If $N(u_2) \setminus V(P)$ is a clique, set $K_2 = N(u_2) \setminus V(P)$, otherwise set $K_2 = \emptyset$.*

Let $Q_P = (K_1 \cup K_2 \cup V(P)) \setminus F_G$. Then $G \setminus F_G$ admits a nice elimination order $v_1, \ldots, v_k$, where Q_P is in the end of this order (i.e. $Q_P = \{v_{k-|Q_P|+1}, \ldots, v_k\}$).

Proof. We prove the lemma when G is a long pyramid or an extended nontrivial basic graph. In fact, since the proof for a long pyramid can be treated almost similarly, we only show here the proof in the case where G is an extended nontrivial basic graph. Suppose that $V(G) = V(H) \cup \{x, y\}$, where H is the line graph of a tree. We may assume the followings:

1. P is a maximal flat path in G (two ends of P are of degree ≥ 3).

   If the lemma is true when P is a maximal flat path then it is also true for all subpaths of P, because Q_P admits a perfect elimination order (an order of vertices in which the neighborhood of a vertex induces a clique at the time it is eliminated) where a fixed subpath of P is in the end of this order.

2. All the vertices in $G \setminus (F_G \cup Q_P \cup \{x, y\})$ are not in C_G.

   Observe that the neighborhood of every vertex v in G, except x and y, induces a union of two cliques. Therefore, if $v \in C_G$, it must have a neighbor of degree 2 in F_G, then its neighborhood in $G \setminus F_G$ actually induces a clique and it can be eliminated at the beginning of our order.

3. Every vertex in $G \setminus (F_G \cup Q_P)$ has at most one neighbor in C_G.

   Indeed, by the assumption 2, $C_G \subseteq \{x, y\} \cup Q_P$. If a vertex $v \in G \setminus (F_G \cup Q_P)$ has two neighbors in C_G, then it must have a neighbor $u \in (C_G \cap Q_P) \setminus \{x, y\}$. By the definition of C_G, v must be a vertex in F_G since it is the only neighbor of degree 2 of u, a contradiction to the choice of v.

Let us first forget about the flat path P and the restriction of putting all the vertices of Q_P in the end of the order. We will show how to obtain a nice elimination order for $G \setminus F_G$ in this case. We choose an arbitrary extended clique K_R in H and call it the *root* clique. For each other extended clique K in H, there exists a vertex $v \in K$ whose removal separates the root clique from $K \setminus v$ in H, we call it *B-vertex*. We call a node *E-vertex* if it is adjacent to x or y. Note that in each extended clique K, we have exactly one B-vertex and at most one E-vertex (by Lemma 3.2.2). For the root clique K_R, we also add a new vertex r adjacent to all the vertices of K_R, and let it be the B-vertex for K_R. Now, if we remove every edge in every extended clique, except the edges incident to its B-vertex, we obtain a tree T_H rooted at r. Note that $V(T_H) = V(H) \cup \{r\}$. We specify the nice elimination order for G where all the vertices in $V(K_R)$ are removed last (we do not care about the order of eliminating r, this vertex is just to define an order for $V(G)$ more conveniently). Let O_T be an order of visiting $V(T_H) \setminus (V(K_R) \cup \{r\})$ satisfying:

- A node u in T_H is visited after all the children of u.
- If u is a B-vertex of some extended clique K, the children of u must be visited in an order where the E-vertex in K (if it exists) is visited last.

Let us introduce some notions with respect to orders first. Let $O_1 = v_1, \ldots, v_k$ and $O_2 = u_1, \ldots, u_t$ be two orders of two distinct sets of vertices. We denote by $O_1 \oplus O_2$ the order $v_1, \ldots, v_k, u_1, \ldots, u_t$. If S is a subset of vertices of some order O_1, we denote by $O_1 \setminus S$ the order obtained from O_1 by removing S. Let u be a vertex, we also denote by u the order of one element u.

Let O_{K_R} be an arbitrary elimination order for the vertices in K_R. Now the elimination order for $G \setminus F_G$ is $O = (O_T \setminus F_G) \oplus x \oplus y \oplus O_{K_R}$. We prove that this elimination order is nice. Indeed, let u be a vertex of order O_T, $u \notin F_G$. If u is not an E-vertex, then its neighborhood at the time it is eliminated is either a subclique of some extended clique (if its parent is a B-vertex) or a single vertex which is its parent. If u is an E-vertex, since it is eliminated after all its siblings (the nodes share the same parent), its neighborhood consists of only two vertices: its parent and x (or y). And by assumption 3, at most one of these two vertices is in C_G, so u is almost simplicial. Now when x is removed in this order, it has at most two neighbors: one is y and one is possibly a vertex in K_R, also not both of them are in C_G, so x is almost simplicial. Vertex y has at most one neighbor at the time it is eliminated. And finally, K_R is a clique so any eliminating order for K_R at this point is nice.

Now we have to consider the flat path P, and put all the vertices of Q_P in the end of the elimination order. There are two cases:

- P is a flat path not containing x and y.

  In this case, both K_1 and K_2 are non-empty. The graph obtained from G by removing $V(P) \cup \{x, y\}$ contains two connected components H_1, H_2, where H_i $(i = 1, 2)$ is the line graph of a tree. By considering K_i as the root clique of H_i, from the above argument, we obtain two elimination orders O_1, O_2 for $H_1 \setminus K_1$ and $H_2 \setminus K_2$. Now, all the vertices not yet eliminated in G are in $\{x, y\} \cup Q_P$. We claim that at least one of x, y has at most two neighbors in the remaining graph. Indeed, otherwise x and y both have at least three neighbors, implying that they both have neighbors in K_1 and K_2, which contradicts Lemma 3.2.2. Suppose x has at most two neighbors, in this case we eliminate x first, then y. Note that x and y are both almost simplicial in this elimination order, since they have at most two neighbors (not both of them in C_G according to assumption 3) at the time they were eliminated. Finally, choose for Q_P a perfect elimination order O_Q. Now the nice elimination order for $G \setminus F_G$ is $O = ((O_1 \oplus O_2) \setminus F_G) \oplus x \oplus y \oplus O_Q$.

- x or y is an end of P.

  W.l.o.g, suppose x is an end of P, say $x = u_1$. In this case $K_1 = \emptyset$, $K_2 \neq \emptyset$. The graph H' obtained by removing $V(P) \cup \{y\}$ from G is the line graph of a tree. We consider K_2 as the root clique of this graph. From above argument, we have a nice elimination order $O_{H'}$ for $V(H') \setminus K_2$. Now all the vertices left in G are in $\{y\} \cup Q_P$. Observe that y has at most two neighbors in the remaining graph (x and possibly a vertex in K_2), therefore y is almost simplicial and can be eliminated. Note that x has no neighbor in K_2, since if $u \in K_2$ is adjacent to x, then $\{u\} \cup N(u)$ is a star cutset in G separating P^* from the rest of the graph (P^* is non-empty since the length of P is at least two). Finally, choose for Q_P a perfect elimination order O_Q. Now the nice elimination order for $G \setminus F_G$ is $O = (O_{H'} \setminus F_G) \oplus y \oplus O_Q$.

□

Proof of Lemma 3.3.2. By Lemma 3.2.6, G has an extreme 2-join decomposition tree $\mathbb{T}_G$. Now, for every node (G', S') of $\mathbb{T}_G$, G' is a special connected even-hole-free graph with no star cutset associated with $(C_{G'}, F_{G'})$ (by Lemmas 3.2.4 and 3.3.1). Now we prove that for every node (G', S') of $\mathbb{T}_G$, G' satisfies Lemma 3.3.2. This implies the correctness of Lemma 3.3.2 since the root of $\mathbb{T}_G$ corresponds to G.

First, we show that for every leaf node (G', S') of $\mathbb{T}_G$, G' satisfies Lemma 3.3.2. If G' is a clique then any elimination order of $G' \setminus F_{G'}$ is nice. If G' is a hole, there exists a vertex v such that one of its neighbors is not in $C_{G'}$, then v can be eliminated first.

The vertices in the remaining graph induce a subgraph of a path, therefore $G' \setminus v$ admits a nice elimination order. If G' is a long pyramid or an extended nontrivial basic graph, we have a nice elimination order for G' by Lemma 3.3.4.

Now, let us prove that Lemma 3.3.2 holds for G', where (G', S') is a non-leaf node of $\mathbb{T}_G$. Since $\mathbb{T}_G$ is extreme, G' admits an extreme 2-join with the split $(X_1, X_2, A_1, B_1, A_2, B_2)$ and let G_1, G_2 be the blocks of decomposition of G' with respect to this 2-join. We may assume that G_1 is basic and G_2 satisfies Lemma 3.3.2 by induction. Note that $V(G') = (V(G_1) \setminus V(P_2)) \cup (V(G_2) \setminus V(P_1))$. Now we try to specify a nice elimination order for G' by combining the orders for G_1 and G_2. Since G_1 is basic, apply Lemma 3.3.4 for G_1 with $P = P_2$, we obtain the nice elimination order O_1 for $G_1 \setminus (F_{G_1} \cup Q_P)$. Remark that all the vertices in O_1 are in $V(G')$ since we have not eliminated Q_P. By induction hypothesis, we obtain also a nice elimination order O_2 for $G_2 \setminus F_{G_2}$. We create an order O_2' from O_2 for $V(G')$ as follows (a_1, b_1 are two ends of the marker path P_1):

- If $a_1 \in C_{G_2}$ and A_1 is a clique, O_2' is obtained from O_2 by substituting a_1 in O_2 by all the vertices in A_1 (in any order), otherwise set $O_2' = O_2 \setminus \{a_1\}$.
- If $b_1 \in C_{G_2}$ and B_1 is a clique, O_2' is obtained from itself by substituting b_1 in O_2' by all the vertices in B_1 (in any order), otherwise set $O_2' = O_2' \setminus \{b_1\}$.

We claim that $O = O_1 \oplus O_2'$ is a nice elimination order for $G' \setminus F_{G'}$. Let $N'_{G'}(u)$ ($N'_{G_1}(u)$, $N'_{G_2}(u)$) be the set of neighbors of u in the remaining graph when it is removed with respect to order O (O_1, O_2, respectively).

- If u is a vertex in O_1.
  - If $u \notin A_1$ and B_1, then $N_{G'}(u) = N_{G_1}(u)$, because u is almost simplicial in G_1 then it is also almost simplicial in G' at the time it was eliminated.
  - If $u \in A_1$ or B_1, w.l.o.g, suppose that $u \in A_1$, then A_1 is not a clique, because we do not eliminate Q_P in O_1. Since one of A_1, A_2 must be a clique to avoid 4-hole, A_2 is a clique. If $a_2 \in F_{G_1}$, then $|A_1| = 1$ and A_1 is a clique of size 1, a contradiction. Then $a_2 \in C_{G_1}$. Because a_2 was not eliminated at the time we remove u in order O_1, $a_2 \in N'_{G_1}(u)$. We can obtain $N'_{G'}(u)$ from $N'_{G_1}(u)$ by substituting a_2 by A_2, therefore u remains almost simplicial in G'.
- If u is a vertex in O_2'.
  - If $u \in X_2 \setminus (A_2 \cup B_2)$, then $N_{G'}(u) = N_{G_2}(u)$, because u is almost simplicial in G_2 then it is also almost simplicial in G' at the time it was eliminated.

- If $u \in A_2$ or B_2, w.l.o.g, suppose $u \in A_2$. We may assume that A_1 is a clique, since otherwise it was eliminated in O_1 before u, implying $N'_{G'}(u) \subseteq N'_{G_2}(u)$ and u is almost simplicial in G'.
  - Suppose $a_1 \in C_{G_2}$. If u is eliminated after a_1, then $N'_{G'}(u) = N'_{G_2}(u)$ and u is almost simplicial in G'. If u is eliminated before a_1, we can obtain $N'_{G'}(u)$ from $N'_{G_2}(u)$ by substituting a_1 by A_1, therefore u remains almost simplicial.
  - Suppose $a_1 \in F_{G_2}$. Since A_1 is a clique and it contains a vertex $v \in F_{G'}$, $|A_1| \leq 2$. If $|A_1| = 2$, then $N_{G'}(v)$ is a clique cutset of size 2 (star cutset) separating v from the rest of G', a contradiction. Thus $A_1 = \{v\}$ and $N'_{G'}(u) = N'_{G_2}(u)$ (since v is the only vertex in A_1 and $v \notin G' \setminus F_{G'}$) and u is almost simplicial in G'.
- If $u \in A_1$ or B_1, w.l.o.g, suppose $u \in A_1$, then A_1 is a clique, since otherwise it was removed in O_1. We can obtain $N'_{G'}(u)$ from $N'_{G_2}(a_1)$ by creating a clique K, which is a subclique of A_1 (K is actually the set of vertices of A_1 going after u in O'_2), and make it complete to $N'_{G_2}(a_2)$. Therefore, u remains almost simplicial in G'.

□

3.3.3 The bound is tight

Now we show how to construct for any $k \geq 3$ an even-hole-free graph G_k with no star cutset such that $\omega(G_k) = k$ and $\chi(G_k) = k + 1$. The set of vertices of G_k: $V(G_k) = A \cup B \cup C \cup D \cup E \cup F$, where A, C, E are cliques of size $(k-1)$; B, D are independent sets of size $(k-1)$ and F is an independent set of size $(k-2)$. The vertices in each set are labeled by the lowercase of the name of that set plus an index, for example $A = \{a_1, \ldots, a_{k-1}\}$. The edges of G_k as follows:

- A is complete to B, C is complete to D.
- b_i is adjacent to c_i, d_i is adjacent to e_i $(i = 1, \ldots, k-1)$.
- a_{k-1} is complete to E.
- d_1 is complete to F.
- a_i is adjacent to f_i $(i = 1, \ldots, k-2)$.

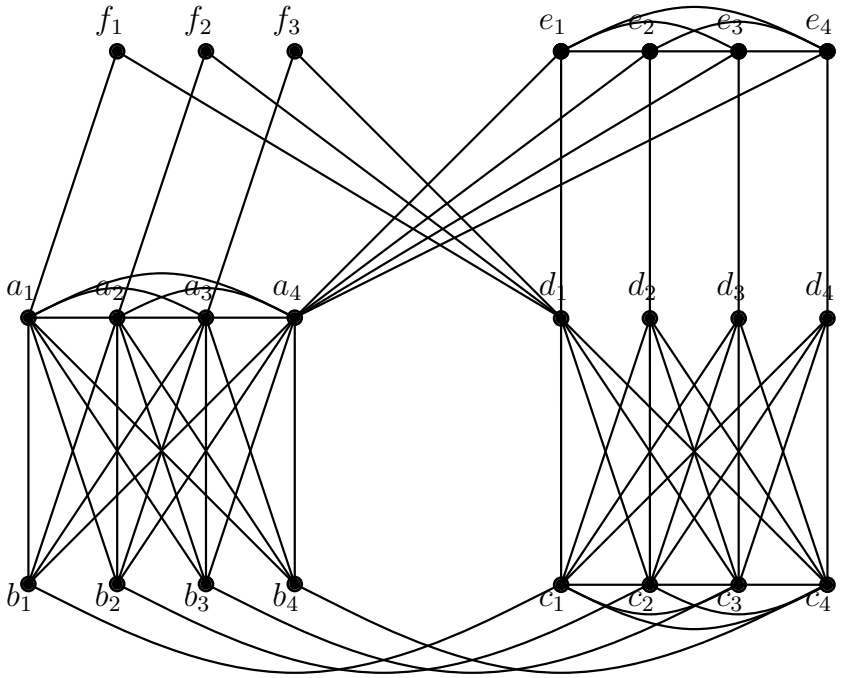


Figure 3.1: Graph G_5 with $\omega(G_5) = 5$ and $\chi(G_5) = 6$

Figure 3.1 is an example of G_k, where $k = 5$. The fact that G_k is an even-hole-free graph with no star cutset can be checked by hand.

Lemma 3.3.5. *For every $k \geq 3$, $\omega(G_k) = k$ and $\chi(G_k) = k + 1$.*

Proof. It is clear that $\omega(G_k) = k$. We will show that G_k is not k-colorable. Suppose we have a k-coloring of G. Because in that coloring, every clique of size k must be colored by all k different colors, then all the vertices in B must receive the same color 1. Therefore, the clique C must be colored by $(k-1)$ left colors, and all the vertices in D must be colored by color 1 also. Therefore, the k-clique $\{a_{k-1}, e_1, \ldots, e_{k-1}\}$ is not colorable since all of them must have color different from 1, a contradiction. □

3.4 Rank-width

3.4.1 Bounded rank-width

Recall that the definition of rank-width and rank-decomposition are given in the last part of Section 3.2. Given a graph G and some rank-decomposition (T, L) of G, a

subset X of $V(G)$ is said to be *separated* in (T, L) if there exists an edge e_X of T corresponding to the partition $(X, V(G) \setminus X)$ of $V(G)$. Let d be an integer, we say that graph G has *property* $\mathbb{P}(d)$ if for every set S of disjoint flat paths of length at least 3 in G, there is a rank-decomposition (T, L) of G such that the width of (T, L) is at most d and every flat path $P \in S$ is separated in (T, L). The next lemma shows the relation between 2-join and rank-width.

Lemma 3.4.1. *Let $\mathbb{D}$ be a class of graphs and $\mathbb{B} \subseteq \mathbb{D}$ be the set of its basic graphs such that every graph $G \in \mathbb{D}$ has a 2-join decomposition tree. Furthermore, there exists an integer $d \geq 2$ such that every basic graph in $\mathbb{D}$ has property $\mathbb{P}(d)$. Then for every graph $G \in \mathbb{D}$, $\mathrm{rwd}(G) \leq d$.*

Proof. Let G be a graph in $\mathbb{D}$ and $\mathbb{T}_G$ be its 2-join decomposition tree. We prove that every node (G', S') of $\mathbb{T}_G$ satisfies the following *property* $\mathbb{P}'(d)$: there is a rank-decomposition (T, L) of G' such that the width of (T, L) is at most d and every flat path $P \in S'$ is separated in (T, L). Note that property $\mathbb{P}'(d)$ is weaker than property $\mathbb{P}(d)$ since it is not required to be true for every choice of the set of disjoint flat paths, but only for a particular set S' associated with G' in $\mathbb{T}_G$. Proving this property for each node in $\mathbb{T}_G$ implies directly the lemma since if the root of $\mathbb{T}_G$ has property $\mathbb{P}'(d)$, then $\mathrm{rwd}(G) \leq d$.

It is clear that every leaf node of $\mathbb{T}_G$ has property $\mathbb{P}'(d)$ since every basic graph has property $\mathbb{P}(d)$ by the assumption. Now we only have to prove that every non-leaf node (G', S') of $\mathbb{T}_G$ has property $\mathbb{P}'(d)$ assuming that its two children (G_1, S_1) and (G_2, S_2) already have property $\mathbb{P}'(d)$. For $i \in \{1, 2\}$, let (T_i, L_i) be the rank-decomposition of G_i satisfying property $\mathbb{P}'(d)$. We show how to construct the rank-decomposition (T, L) of G' satisfying this property. Recall that by the definition of a 2-join decomposition tree, G_1 and G_2 are two blocks of decomposition with respect to some 2-join (X_1, X_2) of G' together with some marker paths $P_2 \in S_1$, $P_1 \in S_2$, respectively. For $i \in \{1, 2\}$, since (G_i, S_i) satisfies property $\mathbb{P}'(d)$, P_{3-i} is separated in (T_i, L_i) by some edge $e_i = u_i v_i$ of T_i. Let C_i, D_i be the two connected components (subtrees) of $T_i \setminus e_i$ (the tree obtained from T_i by removing the edge e_i), where the leaves of C_i correspond to $V(G_i) \setminus V(P_{3-i})$ and the leaves of D_i correspond to $V(P_{3-i})$. W.l.o.g, we may assume that u_i is in C_i and v_i is in D_i. The tree T is then constructed from $T_1[V(C_1) \cup \{v_1\}]$ and $T_2[V(C_2) \cup \{v_2\}]$ by identifying u_1 with v_2 and u_2 with v_1. Note that T is a subcubic tree and the leaves of T now correspond to $V(G)$. The mapping L is the union of the two mappings L_1 and L_2 restricted in X_1 and X_2, respectively. Now the node (G', S') satisfies property $\mathbb{P}'_d$ since:

- Every flat path $P \in S'$ is separated in (T, L).

  It is true since for $i \in \{1, 2\}$, every path $P \in S_i$ is separated in (T_i, L_i).

- The width of (T, L) is at most d.

  It is easy to see that the width of the identified edge $e = u_1v_1$ of T is 2, since it corresponds to the partition (X_1, X_2) of G'. For other edge e of C_i (for $i = 1$ or 2), it corresponds to a cut of G' separating a subset Z of X_i from $V(G')\setminus Z$, and we have $\text{cutrk}_{G'}(Z) = \text{cutrk}_{G_i}(Z)$ (since the rank of the corresponding matrix stays the same if we just add several copies of the columns corresponding to the two ends of the marker path P_{3-i}), which implies that $\text{cutrk}_{G'}(Z) \leq d$.

□

Thanks to Lemma 3.4.1 and the existence of a 2-join decomposition tree by Lemma 3.2.6, to prove that the rank-width of even-hole-free graphs with no star cutset is at most 3, we are left to only prove that every basic even-hole-free graph with no star cutset has property $\mathbb{P}(3)$. Actually, by Observation 3.2.7, we do not have to prove it for cliques and holes, since they never appear in the leaf nodes of any 2-join decomposition tree of any graph in our class. Therefore, Theorem 3.1.2 is a consequence of Observation 3.2.8 and the following lemma:

Lemma 3.4.2. *Every basic even-hole-free graph with no star cutset, which is neither a clique nor a hole, has property* $\mathbb{P}(3)$.

Proof. Let G be a basic even-hole-free graph with no star cutset, which is different from a clique and a hole. Since G is basic and G is neither a clique nor a hole, G must be an extended nontrivial basic graph or a long pyramid. Since the case where G is a long pyramid can be followed easily from the case where it is an extended nontrivial basic graph. We omit the details for long pyramids here.

Let G be an extended nontrivial basic graph, $V(G) = V(H) \cup \{x, y\}$, where H is the line graph of a tree. Let S be some set of flat paths of length at least 3 in G. Now we show how to build the rank-decomposition of G satisfying the lemma.

First, we construct the *characteristic* tree F_H for H. We choose an arbitrary extended clique in H as a *root* clique. Let E be the set of flat paths obtained from H by removing all the edges of every extended clique in H. Now, we define the father-child relation between two flat paths in E. A path B is the *father* of some path B' if they have an endpoint in the same extended clique in H and any vertex of B is a cut-vertex in H which separates B' from the root clique. If B is the father of B' then we also say that B' is a *child* of B. Any path in E which has only one endpoint in an extended clique is called *leaf* path, otherwise it is called *internal* path. Now, we consider each path B in E as a vertex v_B in the characteristic tree F_H, and associate with each node v_B a set $S_{v_B} = V(B)$. Each leaf path corresponds to a leaf

in F_H and each internal path corresponds to an internal node in F_H, which reserves the father-child relation (if a path B is the father of some path B' then v_B is the father of $v_{B'}$ in F_H). We also add a root r for F_H, and the children of r are all the vertices v_B, where B is a path with an endpoint in the root clique, let $S_r = \emptyset$. Now, we add two special vertices x, y to attain the *characteristic* tree F_G for G. If x (or y) is an endpoint of some flat path P in S, then we set $S_v = S_v \cup \{x\}$ ($S_v = S_v \cup \{y\}$, respectively), where v is the leaf in F_H corresponding to the leaf path in E which contains $P \setminus \{x\}$ ($P \setminus \{y\}$, respectively). Otherwise set $S_v = S_v \cup \{x\}$ ($S_v = S_v \cup \{y\}$), where v is a leaf in F_H corresponding to any path in E having an endpoint adjacent to x (y, respectively). Figures 3.2 and 3.3 are the example of an extended nontrivial basic graph G and its characteristic tree F_G (the bold edges are the edges of flat paths in S). Note that each node in F_G corresponds to a subset S_v of $V(G)$, they are all disjoint, each of them induces a flat path in G and $V(G) = \cup_{v \in F_G} S_v$.

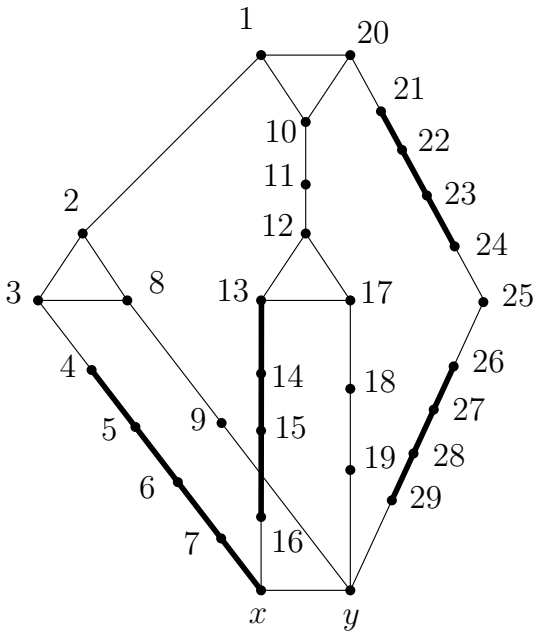


Figure 3.2: An extended nontrivial basic graph G with a set of flat paths.

Now, we show how to build the rank-decomposition of G from its characteristic tree F_G. We first define a special rooted tree, called *k-caterpillar* ($k \geq 1$) to achieve that goal. For $k \geq 1$, a graph I is called *k-caterpillar* if:

- For $k = 1$, $V(I) = \{a_1, l_1\}$, $E(I) = \{a_1 l_1\}$ and a_1 is the root of I.

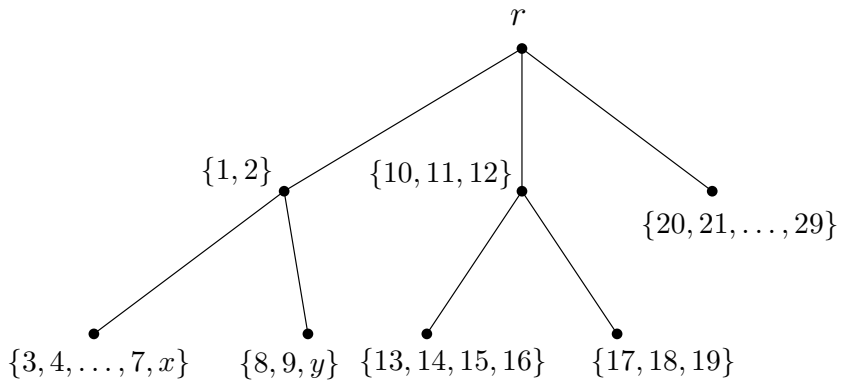


Figure 3.3: The characteristic tree F_G for graph G in Figure 3.2.

- For $k \geq 2$, $V(I) = \{a_1, \ldots, a_{k-1}\} \cup \{l_1, \ldots, l_k\}$, $E(I) = \{a_i a_{i+1} | 1 \leq i \leq k-2\} \cup \{a_i l_i | 1 \leq i \leq k-1\} \cup \{a_{k-1} l_k\}$ and a_1 is the root of I.

Notice that in the following discussion, for the sake of construction, the rank-decomposition (T, L) we build for our graph is not exactly the same as in the definition of a rank-decomposition mentioned in Section 3.2, since we allow vertex of degree 2 in tree T, but it does not change the definition of rank-width. A flat path in G is called *mixed* if it contains a flat path in S but it is not a flat path in S. We start by constructing the rank-decomposition of a non-mixed flat path in G. For a non-mixed flat path $P = p_1 \ldots p_k$, we create a k-caterpillar T_P which has exactly k leaves $l_1, \ldots, l_k$ as in the definition and a bijection L_P maps each vertex in P to a leaf of T_P such that $L_P(p_i) = l_i$. Since a mixed path can always be presented as a union of vertex-disjoint non-mixed paths $P = \cup_{i=1}^{k} P_i$ (where one end of P_i is adjacent to one end of P_{i+1} for $1 \leq i \leq k-1$), let (T_i, L_i) be the rank-decomposition for each non-mixed path P_i constructed as above, we can build the tree T_P by creating a k-caterpillar I which has exactly k leaves $l_1, \ldots, l_k$ as in the definition and identify each root of T_i with the leaf l_i of I for $1 \leq i \leq k$. Also, let the mapping L_P from $V(P)$ to the leaves of T_P be the union of all the mappings L_i's for $1 \leq i \leq k$. Now, we build the rank-decomposition (T_G, L_G) of G from its characteristic tree F_G by visiting each node in F_G in an order where all the children of any internal node is visited before its father. For a vertex $v \in F_G$, denote by C_v the union of all connected components of $F_G \setminus v$ that does not contain r. Let $F_G(v) = F_G[V(C_v) \cup \{v\}]$, $X_v = \cup_{u \in F_G(v)} S_u$. At each node v of F_G, we build the rank-decomposition (T_v, L_v) of the graph G_v induced by the subset X_v of $V(G)$ by induction:

1. If v is a leaf of F_G, build the rank-decomposition (T_v, L_v) for the flat path corresponding to v like above argument for mixed and non-mixed paths.

2. If v is an internal node of F_G different from its root and $v_1, \ldots, v_k$ are its children. Let (T, L) be the rank-decomposition of the flat path corresponding to v (built by above argument for mixed and non-mixed paths) and (T_i, L_i) $(i = 1 \ldots k)$ be the rank-decomposition of $G[X_{v_i}]$. We build T_v by constructing a $(k+1)$-caterpillar having exactly $(k+1)$ leaves $l_1, \ldots, l_{k+1}$ as in the definition and identify the root of T with l_1, the root of T_i with l_{i+1} for $1 \leq i \leq k$. Let the mapping L_v from X_{v_i} to the leaves of T_v be the union of the mapping L and all the mappings L_i's for $1 \leq i \leq k$.

3. If v is the root of F_G and $v_1, \ldots, v_k$ are its children. Let (T_i, L_i) $(i = 1 \ldots k)$ be the rank-decompositions of $G[X_{v_i}]$. We build T_v by constructing a k-caterpillar having exactly k leaves $l_1, \ldots, l_k$ as in the definition and identify the root of T_i with l_i for $1 \leq i \leq k$. Let the mapping L_v from $V(G)$ to the leaves of T_v be the union of all the mappings L_i's for $1 \leq i \leq k$.

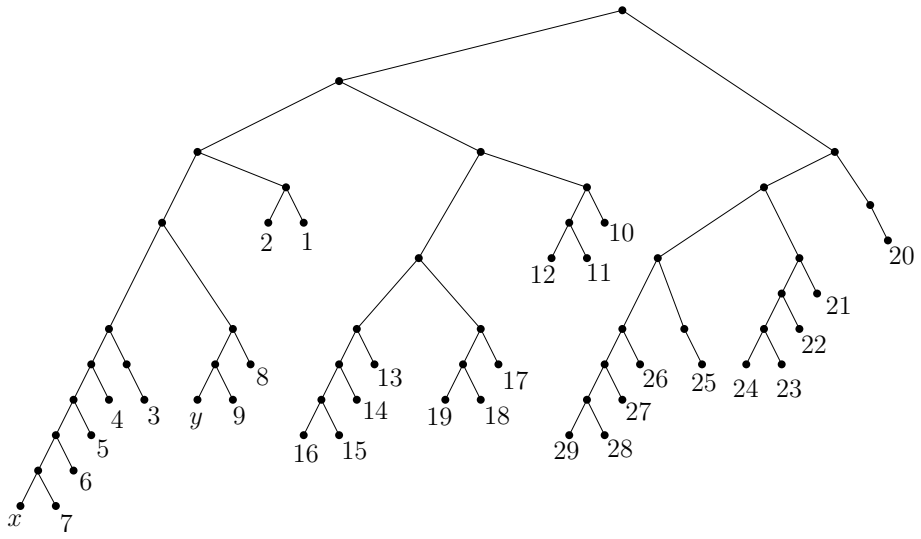


Figure 3.4: The rank-decomposition for graph G in Figure 3.2.

The rank-decomposition (T_r, L_r) corresponding to the root r of F_G is the desired rank-decomposition (T, L) for G (see Figure 3.4). Now we prove that this rank-decomposition construction for the extended nontrivial basic graphs G satisfies the lemma.

Proposition 3.4.3. *Let (T, L) be the above constructed rank-decomposition for G. Then, every flat path P in S is separated in (T, L).*

Proof. It is trivially true, because P is a non-mixed subpath of some flat path B in E, so $V(P)$ is separated in the rank-decomposition of B. And each flat path B of G is also separated in the rank-decomposition of G by our construction. So $V(P)$ is separated in (T, L). □

Proposition 3.4.4. *The above constructed rank-decomposition (T, L) of G has width at most 3.*

Proof. We prove by the structure of the characteristic tree F_G of G. For an internal node v of F_G, let $v_1, \ldots, v_k$ be its children, in some sense, the decomposition tree T_v for X_v is obtained by "glueing" the decomposition tree for $G[S_v]$ and all the decomposition trees T_i for $G[X_{v_i}]$ for $1 \leq i \leq k$ along a cut-vertex. Therefore, we consider an edge e of T_v as an edge of T as well. Our goal is to prove that the width of any edge e with respect to the rank-decomposition (T, L) of G is at most 3. For the sake of induction, at each node v of F_G, we prove that the width of any edge e of T_v is at most 3 with respect to the rank-decomposition (T, L) of G (we mention v here just to specify an edge in our tree T):

1. If v is a leaf in F_G. Every edge e of T_v corresponds to a partition of $V(G)$ into two parts where one of them is a subpath of the flat path corresponding to v, so the width of e is at most 2.

2. If v is an internal node in F_G and $v_1, \ldots, v_k$ are its children. Let (T_i, L_i) $(i = 1 \ldots k)$ be the rank-decompositions of $G[X_{v_i}]$. Let e be an edge of T_v. If e is an edge of T_i then the width of e is at most 3 by induction. Otherwise, e corresponds to one of the following situations:

   - e corresponds to a partition $(V(P), V(G) \setminus V(P))$ of $V(G)$, where P is a subpath of the flat path $G[S_v]$. In this case, the width of e is clearly at most 2.
   - e corresponds to a partition $(U, V(G) \setminus U)$ of $V(G)$, where U is the union of several X_{v_i}'s. Let K be the extended clique intersecting every X_{v_i}. In this case, there are only three types of neighborhood of vertices of U in $G \setminus U$:
     - $K \setminus U$,
     - x if $x \notin U$, or $N(x) \setminus U$ if $x \in U$, and
     - y if $y \notin U$, or $N(y) \setminus U$ if $y \in U$.

     Therefore, the width of e is at most 3.

□

Lemma 3.4.2 is true because of the Propositions 3.4.3 and 3.4.4. □

3.4.2 An even-hole-free graph with no clique cutset and unbounded rank-width

It is clear that clique cutset is a particular type of star cutset. However, the class of even-hole-free graph with no clique cutset (a super class of even-hole-free graph with no star cutset) does not have bounded rank-width. Since clique-width and rank-width are equivalent, now we show how to construct for every $k \geq 4$, k even an even-hole-free graph G_k with no clique cutset and $\text{cwd}(G_k) \geq k$. The set of vertices of G_k: $V(G_k) = \cup_{i=0}^{k} A_i$, where each $A_i = \{a_{i,0}, \ldots, a_{i,k}\}$ is a clique of size $(k+1)$. We also have edges between two consecutive sets A_i, A_{i+1} $(i = 0, \ldots, k$, the indexes are taken modulo $(k+1))$. They are defined as follows: $a_{i,j}$ is adjacent to $a_{i+1,l}$ iff $j + l \leq k$.

Lemma 3.4.5. *For every $k \geq 4$, k even, G_k is an even-hole-free graph with no clique cutset.*

Proof. By the construction, there is no hole in G_k that contains two vertices in some set A_i and every hole must contain at least a vertex in each set A_i. Therefore, every hole in G_k has exactly one vertex from each set A_i, so its length is $(k+1)$ (an odd number). Hence, G_k is even-hole-free.

We see that every clique in G_k is contained in the union of some two consecutive sets A_i, A_{i+1}. Hence, its removal does not disconnect G_k. Therefore, G_k has no clique cutset. □

Lemma 3.4.6. *For every $k \geq 4$, k even,* $\text{cwd}(G_k) \geq k$.

Proof. The graph obtained from G_k by deleting all the vertices in $A_0 \cup_{i=1}^{k} \{a_{i,0}\}$ is isomorphic to the permutation graph H_k introduced in [27]. And because it was already proved in that paper that $\text{cwd}(H_k) \geq k$, and clique-width of G_k is at least the clique-width of any of its induced subgraph then $\text{cwd}(G_k) \geq k$. □

Note that an example of a (diamond, even-hole)-free graph with no clique cuset and unbounded rank-width was also given in [50] (a *diamond* is a graph obtained from a complete graph on four vertices by removing one edge). This graph is constructed differently from ours. It is built by specifying the edges between a long path and a large clique in such a way that there is no even hole. However, our construction

and theirs both contain a large clique. We do not know whether this is the case for even-hole-free graphs in general. The following question was asked in [8]:
Open question. Is the rank-width of an even-hole-free graph bounded by a function of its clique number?

Chapter 4

Connected greedy coloring

A natural way to color a graph is by using a greedy algorithm: we consider the vertices of a graph in sequence and assign for each vertex the first available color. Although this algorithm does not always give the optimal solution, it is very practical and may give some information about the structure of a graph. Many researches have been done for general orders (see [10, 21, 52, 7]): the complexity of computing Grundy number (i.e. maximum k such that there exists an order producing a k-coloring), the characterization of graphs where there exists a bad ordering, the complexity of recognizing (hereditary) well-colored graphs, ... There are many ways to choose an order of the vertices to hopefully improve the outcome of greedy algorithms. In this chapter, we mainly focus on *connected* orders, an order where each vertex (except the first one) has a neighbor before it in the order. Connected orders have also been studied [31, 2, 4, 7]. We know that it is NP-hard to compute connected Grundy number (a similar parameter for connected order). Some examples of graphs that are not friendly with connected orders were also given. However, we know very little about the characterization of good graphs with respect to connected orders. A graph G is *good* if for every connected induced subgraph H of G and for every connected order $\mathcal{O}$ of H, the greedy algorithm gives H an optimal coloring. In this chapter, we give the characterization of good claw-free graphs in terms of minimal forbidden induced subgraphs. This also implies an algorithm for recognizing good claw-free graphs. Note that the complexity of recognizing good graphs remains open. The result of this chapter is covered in the following paper:

[IV] N.K. Le, N. Trotignon. Connected greedy colouring in claw-free graphs, *arXiv preprint*, submitted.

4.1 Introduction

Let G be a graph and $\mathcal{O} = [v_1, \ldots, v_n]$ be a linear ordering of its vertices. The *greedy coloring algorithm* (greedy algorithm for short) applied to $(G, \mathcal{O})$ consists in taking the vertices in the order $\mathcal{O}$, and giving to each vertex a color equal to the smallest positive integer not used by its neighbours already colored. This obviously produces a coloring.

For every graph, there exists an order $\mathcal{O}$ for the vertices such that the greedy algorithm produces an optimal coloring. To see this, consider an optimal coloring π, and consider the following ordering: first take vertices with color 1, then vertices with color 2, and so on. But this method has no practical interest to compute optimal colorings, since to find the ordering, an optimal coloring has to be known.

It is also well known that for some graphs, there exist orderings that produce colorings very far from the optimal, for instance consider two disjoint sets on n vertices, say $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$. Add all possible edges between A and B, except edges $a_i b_i$, $i \in \{1, \ldots, n\}$. This produces a bipartite graph G. However, the greedy algorithm applied to the order $[a_1, b_1, a_2, b_2, \ldots, a_n, b_n]$ produces a coloring with n colors.

One might wonder for which graphs the greedy algorithm always gives an optimal solution no matter what order is given. The operation *Disjoint-Union* consists in building a new graph by taking the union of two vertex-disjoint graphs. The operation *Complete-Join* consists in building a new graph by taking the union of two vertex-disjoint graphs G_1 and G_2, and by adding all possible edges between $V(G_1)$ and $V(G_2)$. Let P_k denote the path on k vertices. A *cograph* is a P_4-free graph. Seinsche [46] proved that cographs are exactly the graphs that can be produced by starting with graphs on one vertex and by repeatedly apply the operations Disjoint-Union and Complete-Join to previously constructed graphs. The graphs such that the greedy algorithm on every order gives every induced subgraph of them an optimal coloring are fully characterized.

Theorem 4.1.1 (see [21, 10])**.** *For every graph G, the following properties are equivalent.*

- *G is a cograph.*
- *For every induced subgraph H of G and every linear order $\mathcal{O}$ of $V(H)$, the greedy coloring algorithms applied to $(H, \mathcal{O})$ produces an optimal coloring of H.*

There are many ways to order the vertices of a graph with the hope to obtain a better coloring. In this paper, we focus on *connected* orders. An order $\mathcal{O} =$

$[v_1, \ldots, v_n]$ for a graph G is *connected* if for every $2 \leq i \leq n$, there exists $j < i$ such that $v_j v_i \in E(G)$. A connected order exists if and only if G is connected, and is efficiently produced by search algorithms such as BFS, DFS (or more simply by the algorithm *generic search*). We say that a graph G is *good* if for every connected induced subgraph H of G and every connected order $\mathcal{O}$ of H, the greedy algorithm produces an optimal coloring of H. Also, a connected order $\mathcal{O}$ of a graph G is *good* if it produces an optimal coloring of G. A graph or a connected order is *bad* if it is not good. A graph is *minimally bad* if it is bad and all other connected induced subgraphs of it are good. Connected orders are better than general orders for coloring bipartite graphs.

Theorem 4.1.2 (see [4])**.** *Every bipartite graph is good.*

However, unlike general orders, it is not true that for every graph, there exists a connected order that provides an optimal coloring, see [2] for example. A similar claw-free example is given in Figure 4.1.

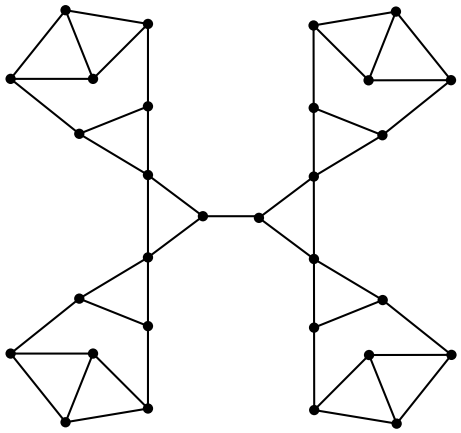


Figure 4.1: A claw-free graph where every connected order is bad.

The connected greedy coloring has recently been studied. In [4], they define $\Gamma_c(G)$ (also known as *connected Grundy number*) as the maximum number k such that there exists a connected order producing a k-coloring of G. They also proved that checking if $\Gamma_c(G) \geq k$ is NP-hard if k is a part of the input. In [7], they show that this problem remains NP-hard even when $k = 7$. A graph G is good in our definition if for every connected induced subgraph H of G, $\Gamma_c(H) = \chi(H)$. Note that their results imply also that checking if there exists a bad connected order for a graph is NP-hard, but do not imply NP-hardness on recognizing good graphs (since a class of good graphs is hereditary by our definition). The complexity of recognizing

good graphs remains open. In [2], they gave several examples of small graphs that are not friendly with connected orders. They also proved that gem (see Figure 4.2) is the unique smallest bad graph. In [31], they defined a more restricted good graph with respect to connected orders and gave the complete characterization of this class. Therefore, their class is also good by our definition.

However, the list of excluded induced subgraphs for the class of good graphs is still unknown. Equivently, no description of minimally bad graphs is known. Our goal is to prove an analogue of Theorem 4.1.1 for connected orders. If we restrict our attention to claw-free graphs, we are able to give this description. This is our main result that we now state precisely. The rest of the chapter is devoted to its proof.

The main result

Note that the definition of a prism is presented in Section 1.4 of Chapter 1. Here we add the definitions of some particular types of prism. A prism is *short* if one of its three paths is of length 1. A prism is *parity* if its three paths have the same parity and is *imparity* otherwise. Note that a prism contains an odd hole if and only if it is imparity. A parity prism is *even* (*odd*) if the lengths of its three paths are even (odd).

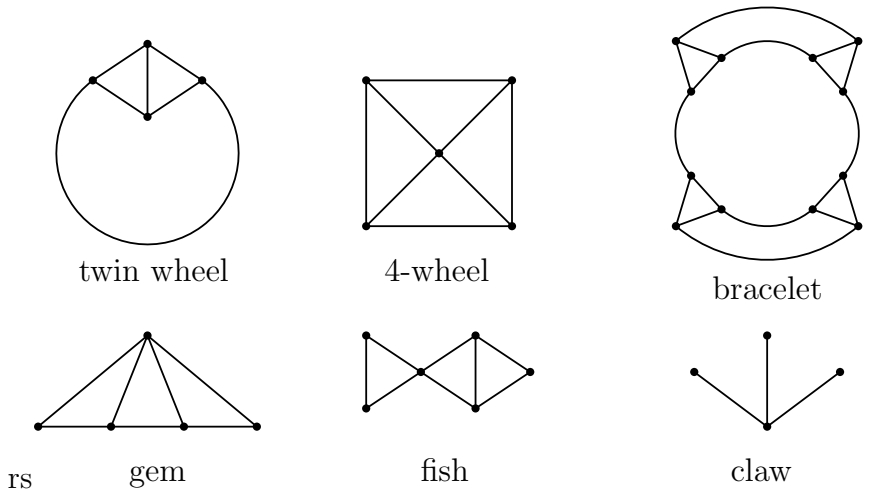


Figure 4.2: Some graphs

We also need several particular graphs, defined in Figure 4.2. We call *obstructions* the graphs represented in Figure 4.3. Note that in Figures 4.2 and 4.3, all the straight lines are edges, all the curved lines are paths of length ≥ 1. In Figure 4.2:

- The path of a twin wheel is of length at least 2.
- A bracelet has six paths of length at least 2: two paths in the sides are of even length; the other four paths are of odd length.

The graphs in Figure 4.3 have the following specifications:

- The orientation represented for each graph has no special meaning. It is an indication of how a bad connected order can be found for it. The orientation does not fully specify this order. The arrow should be seen from a small to a big vertex with respect to this order. The chromatic number of each graph is 3 and the last vertex in every bad order receives color 4.
- The hole in F_1 is odd.
- The only path in F_2 is of length ≥ 1. The orientation of the only unoriented edge depends on the parity of this path. F_2 is a gem when the length of this path is 1.
- The only path in F_3 is of length ≥ 1.
- The hole in F_5 is even.
- All paths in F_7, F_8, F_9, F_{10} are of length ≥ 2.
- F_7 is an imparity prism. The lower path is of different parity from the other two paths.
- The prism in F_8 is an even prism. The upper path of the prism contains two flat paths: the first one is odd, the second is even.
- The prisms in F_9 and F_{10} are odd prisms.
- The upper path of the prism in F_9 contains two odd flat paths.
- The upper and lower paths of the prism in F_{10} contain four even flat paths.
- The length of the only long cycle in F_{11} is odd ≥ 3. If its length is 3, then F_{11} is a fish.
- The length of two flat paths in F_{12} is odd ≥ 3.

Our main result is the following.

Theorem 4.1.3. *Let G be a claw-free graph. Then G is good if and only if G does not contain any obstruction as an induced subgraph. Equivalently, a claw-free graph is minimally bad if and only if it is an obstruction.*

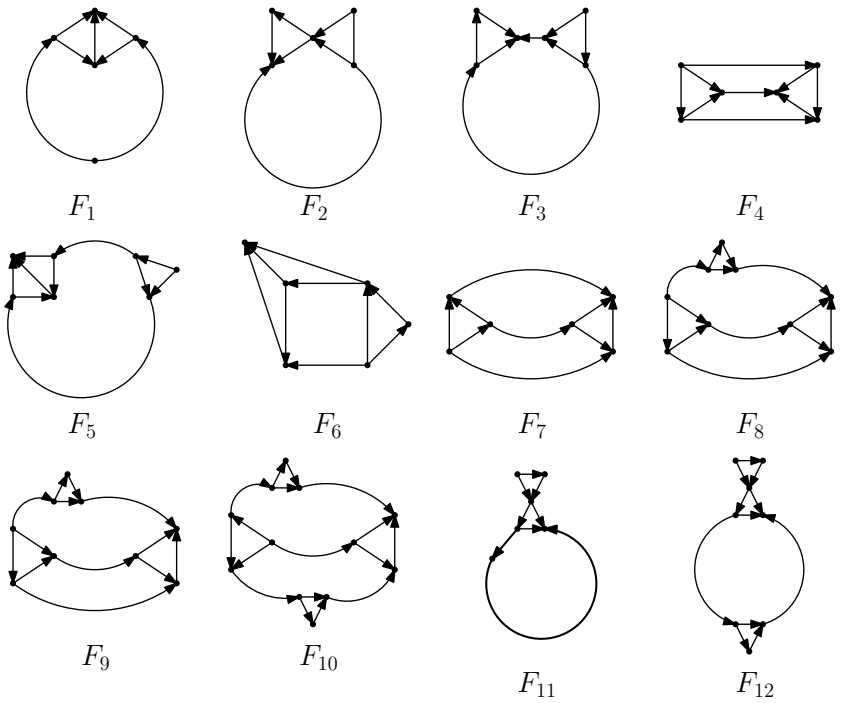


Figure 4.3: List of obstructions

4.2 Some properties of minimally bad graphs

For any graph G, any order $\mathcal{O}$ of its vertices and any vertex v, let $\pi_{G,\mathcal{O}}(v)$ be the color that vertex v receives when applying the greedy coloring algorithm to G with order $\mathcal{O}$. We also write $\pi(v)$ or $\pi_{\mathcal{O}}(v)$ when the context is clear.

Let G be a graph with an ordering $\mathcal{O} = [v_1, \ldots, v_n]$ of its vertices. For vertices u, v of G, we use the notations $u <_{\mathcal{O}} v$, $u >_{\mathcal{O}} v$, $u \leq_{\mathcal{O}} v$, $u \geq_{\mathcal{O}} v$ with the obvious meaning. When clear from the context, we omit the subscript $\mathcal{O}$. When v is a vertex of G, we denote by $G_{\leq v}$ the subgraph of G induced by $\{u \in V(G) \text{ such that } u \leq v\}$. Similarly, we use the notations $G_{<v}$, $G_{\geq v}$ and $G_{>v}$.

When $X \subseteq V(G)$, we use the notation $\mathcal{O}[X]$ to denote the order induced by $\mathcal{O}$ on X, and $\mathcal{O} \setminus X$ to denote the order induced by $\mathcal{O}$ on $V(G) \setminus X$. We write $\mathcal{O} \setminus v$ instead of $\mathcal{O} \setminus \{v\}$. We denote by $\max(X)$ (resp. $\min(X)$) the maximum (resp. minimum) element in X.

Let G be a graph and $\mathcal{O} = [v_1, \ldots, v_n]$ be a linear ordering of its vertices. The *greedy coloring algorithm starting with color 2* applied to $(G, \mathcal{O})$ consists in giving v_1 color 2, and then taking the vertices from v_2 on in the order $\mathcal{O}$, and giving to

each vertex a color equal to the smallest positive integer not used by its neighbours already colored.

Lemma 4.2.1. *When applied to a good graph, the greedy coloring algorithm starting with color 2 produces an optimal coloring.*

Proof. The coloring produced by this algorithm is the same as the coloring produced by the connected order obtained from $\mathcal{O}$ by swapping the first two vertices. Hence it is optimal. □

For the rest of this section, G is a minimally bad graph with a bad order $\mathcal{O} = [v_1, \ldots, v_n]$. Note that for any set $S \subsetneq V(G)$, if $\mathcal{O}[S]$ is a connected order then it produces an optimal coloring for $G[S]$.

Lemma 4.2.2. *For every $x \in V(G) \setminus \{v_n\}$, $\pi(x) \leq \chi(G)$ and $\pi(v_n) = \chi(G) + 1$.*

Proof. Follows directly from the fact that $\mathcal{O}$ is a bad order and that G is a minimally bad graph. □

Lemma 4.2.3. *If $x \in V(G) \setminus \{v_n\}$ and $\mathcal{O} \setminus x$ is connected, then for some vertex $y \neq x$ in G, $\pi_{G\setminus x, \mathcal{O}\setminus x}(y) \neq \pi_{G,\mathcal{O}}(y)$.*

Proof. The conclusion is true for $y = v_n$. Because by the minimality of G, we have $\pi_{G\setminus x, \mathcal{O}\setminus x}(v_n) \leq \chi(G \setminus x) \leq \chi(G)$ and by Lemma 4.2.2, $\pi(v_n) = \chi(G) + 1$. □

Lemma 4.2.4. $\pi(v_n) \geq 4$.

Proof. Otherwise, $\pi(v_n) \leq 3$, so by Lemma 4.2.2, $\chi(G) \leq 2$, so G is bipartite, a contradiction to Theorem 4.1.2. □

Lemma 4.2.5. *For every vertex $v \in V(G)$, $G_{\leq v}$, $G_{\geq v}$, $G_{<v}$ and $G_{>v}$ are connected. In particular, G is connected.*

Proof. For $G_{\leq v}$, it comes from the definition of connected orders.

Suppose $C_1, \ldots, C_k$ ($k \geq 2$) are the connected components of $G_{\geq v}$. For $i = 1, \ldots k$, set $G_i = G[\{u \in V(G) \text{ such that } u < v\} \cup C_i]$ and let $\mathcal{O}_i$ be the order $\mathcal{O}$ restricted to $V(G_i)$. For every $i \in \{1, \ldots, k\}$ and for every vertex $u \in C_i$, we have $\pi_{G,\mathcal{O}}(u) = \pi_{G_i,\mathcal{O}_i}(u)$ because there are no edges in G between C_i and C_j for $i \neq j$. But since G is minimally bad, $V(G_i) \subsetneq V(G)$ and $\mathcal{O}_i$ is a connected order, $\pi_{\mathcal{O}_i}$ is an optimal coloring for G_i. So, for every vertex u in G, $\pi(u) \leq \chi(G_i) \leq \chi(G)$, so π is an optimal coloring, a contradiction.

The proof is the same for $G_{<v}$ and $G_{>v}$ (note that we view the empty graph as a connected graph). □

Lemma 4.2.6. *If S is a cutset of G, then for every component C of $G \setminus S$ except at most one,* $\max(C) < \max(S)$. *Furthermore, if C is the unique component such that* $\max(C) > \max(S)$, *then $v_n \in C$.*

Proof. For the first claim, if $\max(C) > \max(S)$ for more than one component C, then $G_{>\max(S)}$ is disconnected, a contradiction to Lemma 4.2.5. The second claim follows trivially. □

Lemma 4.2.7. *Suppose S is a clique cutset of G and C is a component of $G \setminus S$ such that* $\max(S) < \min(C) = v$. *If v is complete to S, then there exists $u \in S \cup \{v\}$ such that $\pi(u) > |S| + 1$.*

Proof. Otherwise, since $S \cup \{v\}$ is a clique, the colors $1, \ldots, |S| + 1$ are exactly the colors used in $S \cup \{v\}$. Now build an order $\mathcal{O}'$ of $G[S \cup C]$ by first reordering the vertices from $S \cup \{v\}$ by increasing order of their colors, and then taking the rest of $S \cup C$ as it is ordered by $\mathcal{O}$. This new order is connected (as $\mathcal{O}$) and therefore provides an optimal coloring of $G[S \cup C]$. It also gives the same coloring as $\mathcal{O}$ for $G[S \cup C]$. Since by Lemma 4.2.6 $v_n \in C$, it follows that $\pi(v_n) \leq \chi(G[S \cup C]) \leq \chi(G)$, a contradiction to Lemma 4.2.2. □

Lemma 4.2.8. *For $v \in V(G)$, let S be a cutset of $G_{\leq v}$. If there exists a connected component C of $G_{\leq v} \setminus S$ such that* $\min(C) < \min(S)$ *then $v_1 \in C$.*

Proof. If $v_1 \notin C$, then $G_{<\min(S)}$ is not connected: v_1 and $\min(C)$ are in different components, a contradiction to Lemma 4.2.5. □

It is sometimes convenient to view G and $\mathcal{O}$ as an oriented graph D_G, obtained from G by orienting from u to v every edge uv such that $u < v$. We therefore use the notion of *in-neighbor*, *outneighbor*, *source* and *sink* in G (a source in G is a vertex with no in-neighbor in D_G and a sink in G is a vertex with no outneighbor in D_G).

Lemma 4.2.9. *G has a unique source that is v_1 and a unique sink that is v_n.*

Proof. Obviously, v_1 is a source and v_n is a sink. If G has two sources $u < v$, then $G_{\leq v}$ is disconnected (u and v are in two distinct components), a contradiction to Lemma 4.2.5. If G has two sinks $u < v$, then $G_{\geq u}$ is disconnected (u and v are in two distinct components), a contradiction to Lemma 4.2.5. □

Lemma 4.2.10. *Let v be a vertex of degree 2 in G and let $b < a$ be its neighbors. One and exactly one of the following outcome occurs:*

- *$v = v_1$ is the source of G and $v_2 = b$;*

- $b < v < a$.

Moreover, $\pi(v) \in \{1, 2\}$.

Proof. If $b < a < v$, then v is a sink of G and $v = v_n$ by Lemma 4.2.9. Since v has degree 2, $\pi(v) \leq 3$, a contradiction to Lemma 4.2.4.

If $v < b < a$ then v is a source of G and $v = v_1$ by Lemma 4.2.9. Hence, $\pi(v) = 1$. Also, $v_2 = b$ because $\mathcal{O}$ is connected.

Otherwise, $b < v < a$. So, v has degree 1 in $G_{\leq v}$ and $\pi(v) \in \{1, 2\}$. □

Lemma 4.2.11. *In coloring* π*, the colors of the internal vertices of any flat path in* G *alternates between* 1 *and* 2*.*

Proof. Clear by Lemma 4.2.10. □

Lemma 4.2.12. *If* P *is a flat path of* G*, then* $\max(V(P))$ *is an end of* P*.*

Proof. If P has length at most 1, the conclusion is trivial. Otherwise, the ends of P form a cutset of G (note that $G = P$ is impossible since a path is a good graph by Theorem 4.1.2). If $\max(V(P))$ is not an end of P, then by Lemma 4.2.6, v_n is an internal vertex of P. So, by Lemma 4.2.11, $\pi(v_n) \in \{1, 2\}$, a contradiction to Lemma 4.2.4. □

A path $P = p_1 \dots p_k$ in G is *well ordered* if $p_1 < p_2 < \cdots < p_k$ or $p_k < \cdots < p_2 < p_1$. A flat path in G is *maximal* if its two end are not of degree 2 in G.

Lemma 4.2.13. *If* $P = a \dots b$ *is a flat path in* G *then either it is well ordered, or the source* v_1 *is an internal vertex of* P *and* aPv_1*,* v_1Pb *are both well ordered. In particular, there exists at most one maximal flat path in* G *that is not well ordered.*

Proof. This follows from Lemmas 4.2.10 and the definition of connected orders. □

Lemma 4.2.14. *Let* $k \geq 2$ *and* $S = \{s_1, \dots, s_k\}$ *be a set of vertices in* G *such that* $s_1 < \cdots < s_k$ *and* s_k *is complete to* $\{s_1, \dots, s_{k-1}\}$*. Let* $a_1, \dots, a_k \in G \setminus S$ *be* k *distinct vertices of degree* 2 *in* G *and such that* $N_S(a_i) = \{s_i\}$*. Suppose that for* $i = 1, \dots, k-1$*,* $N(s_i) \setminus \{a_i, s_k\} \subseteq N(s_k)$*. If* $a_k < s_k$ *then:*

(1) For every $v \in N(s_k) \setminus \{a_k\}$ *such that* $v < s_k$*,* $\pi(v) \neq \pi(a_k)$*.*

(2) $\pi(a_1) = \dots = \pi(a_k) = 1$ *or* $\pi(a_1) = \dots = \pi(a_k) = 2$*. In particular,* $\{a_1, \dots, a_k\}$ *is a stable set of* G*.*

Proof. To prove (1), suppose that there exists a vertex $v \in N(s_k) \setminus \{a_k\}$ such that $v < s_k$ and $\pi(v) = \pi(a_k)$. Let $b \neq s_k$ be the second neighbor of a_k. Since $a_k < s_k$, by Lemma 4.2.10, a_k is the source of G, or $b < a_k < s_k$. In either case, we can see that $\mathcal{O} \setminus a_k$ is a connected order for $G \setminus a_k$, because $s_{k-1} < s_k$ (and $k \geq 2$).

If a_k is not the source of G, order $\mathcal{O} \setminus a_k$ gives an optimal coloring π' of $G \setminus a_k$ because G is minimally bad. Morevover, for every vertex $u \neq a_k$ in G, we have $\pi'(u) = \pi(u)$. For $u = b$ this is because $b < a_k$, for the other $u < s_k$ this is because a_k brings no constraint to u and for $u = s_k$, this is because the only constraint brought by a_k is also brought by v (because $\pi(v) = \pi(a_k)$). So, π is an optimal coloring of G, a contradiction.

If a_k is the source of G, then $b = v_2$ because $s_{k-1} < s_k$. Hence, $\pi(b) = 2$. We consider the greedy algorithm starting with color 2 applied to $(G \setminus a_k, \mathcal{O} \setminus a_k)$. This is a connected order, and it therefore provides an optimal coloring π' of $G \setminus a_k$ by Lemma 4.2.1. Again, for every vertex of G, $u \neq a_k$, we have $\pi'(u) = \pi(u)$, because the only constraint brought by a_k is given to s_k, and v gives the same constraint. So, π is an optimal coloring of G, a contradiction.

Let us now prove (2). By Lemma 4.2.10, we know that for $i = 1, \ldots, k$, $\pi(a_i) = 1$ or $\pi(a_i) = 2$. If $\pi(a_k) = 1$, then suppose that for some $i < k$, $\pi(a_i) = 2$. No neighbor of s_i smaller than s_i has color 1: for a_i by assumption, and all others are in $N(s_k) \setminus \{a_k\}$, so we know this by (1). Hence, $\pi(s_i) = 1$, contradicting (1). If $\pi(a_k) = 2$, the proof is similar. □

Lemma 4.2.15. *Suppose that G is claw-free. Let s_1, s_2 be two vertices in G such that $s_1 < s_2$ and $s_1 s_2 \in E(G)$. Let a_1, a_2 be distinct vertices of degree 2 in G, such that $a_1 s_1, a_2 s_2 \in E(G)$ and $a_2 < s_2$. Suppose that $N(s_1) \setminus \{a_1, s_2\} = N(s_2) \setminus \{a_2, s_1\} = K$, where K is a non-empty clique. Suppose that $\{s_1, s_2\}$ is a cutset in G and C_1, C_2 are two connected components of $G \setminus \{s_1, s_2\}$ such that $a_1, a_2 \in C_1$ and $K \subseteq C_2$.*

So, $\pi(a_1) = \pi(a_2) = 2$, $s_1 < a_1$ and there exist vertices $v \in K$, $p, q \in C_2 \setminus K$ such that vpq is a triangle, $v < s_1$ and $v < s_2$.

Proof. We first prove that $v = \min(K) < s_2$. Otherwise, $s_2 < v$. Also, $v = \min(C_2)$ because $\mathcal{O}$ is connected. In $G_{\leq s_2}$, s_1 and s_2 both have degree at most 2, so $\pi(s_1), \pi(s_2) \in \{1, 2, 3\}$. In $G_{\leq v}$, v has degree 2, so $\pi(v) \in \{1, 2, 3\}$. Hence, the clique cutset $S = \{s_1, s_2\}$, C_2 and v contradict Lemma 4.2.7. This proves our claim.

By Lemma 4.2.14, we consider two cases.

Case 1: $\pi(a_1) = \pi(a_2) = 1$.

By Lemma 4.2.14, $\pi(v) \neq 1$. So, there exists x adjacent to v with $x < v$ and $\pi(x) = 1$. Note that $x \notin K$ because $x < v$ and $x \notin \{s_1, s_2\}$ because $\pi(x) = 1$. If $s_1 < v$, then $G_{<v}$ is disconnected (x and s_1 are in different components). Therefore,

$v < s_1$. We then have $s_1 < a_1$ for otherwise, $G_{<s_1}$ is disconnected (a_1 and v are in different components). So, $\pi(s_1) = 1$ (since no vertex smaller than s_1 in K has color 1 by Lemma 4.2.14). This is a contradiction because $\pi(a_1) = 1$.

Case 2: $\pi(a_1) = \pi(a_2) = 2$.

First, $\pi(s_1) = 1$ since if $\pi(s_1) \geq 2$, there exists a vertex $u \in K$ such that $u < s_1$ and $\pi(u) = 1$. So, $s_1 < a_1$ for otherwise $G_{<s_1}$ is disconnected. Since by Lemma 4.2.14 no vertex smaller than s_1 in K receives color 2, s_1 receives color 2, a contradiction.

By Lemma 4.2.14, $\pi(v) \neq 2$ and because of s_1, $\pi(v) \neq 1$. So, $\pi(v) \geq 3$. Hence, some in-neighbor q of v ($q \notin K$) satisfies $\pi(q) = 2$. If $s_1 < v$, then $G_{<v}$ is disconnected (s_1 and q are in different components). So, $v < s_1$. Therefore, v must have an in-neighbor $p \notin K$, with $\pi(p) = 1$. Now, $pq \in E(G)$ since G is claw-free. Finally, $s_1 < a_1$, for otherwise $G_{<s_1}$ is disconnected (v and a_1 are in different components). □

4.3 Forbidden structures of minimally bad graphs

Throughout this section, let G be a minimally bad claw-free graph that is not an obstruction.

A graph H is a *cap* in G if:

- $V(H) = K \cup V(P)$.
- K is a clique disjoint from P, $K = L \cup R \cup C$ such that L, R, C are non-empty.
- P is a flat path in G of odd length ≥ 1 with two ends a, b.
- a is complete to L, b is complete to R.
- These are the only edges in H.
- No vertex in $L \cup R \cup V(P)$ has a neighbor in $G \setminus H$.

Lemma 4.3.1. *G does not contain a cap.*

Proof. Suppose G contains a cap H and K, L, R, C, P, a, b are defined as in the definition of a cap. Let a' and b' be the vertices adjacent to a and b in P, respectively. By Lemma 4.2.12, we may assume up to symmetry that $b = \max(V(P))$.

Claim 4.3.2. *For every vertex $v \in K$, if $v < b$, then $\pi(v) \neq \pi(b)$.*

Otherwise, there exists $v < b$ such that $\pi(v) = \pi(b)$. Note in particular that by Lemma 4.2.2, $b \neq v_n$. Also, the existence of v implies that $\mathcal{O} \setminus b$ is a connected order for $G \setminus b$. We then see that for every vertex $y \neq b$, $\pi_{G\setminus b, \mathcal{O}\setminus b}(y) = \pi_{G,\mathcal{O}}(y)$, a contradiction to Lemma 4.2.3. This proves Claim 4.3.2.

Claim 4.3.3. *If $\pi(a) = 1$ or $\pi(a) = 2$, then every vertex $v \in K$ satisfies $v > b$.*

Since $\pi(a) = 1$ or $\pi(a) = 2$, we have $\pi(b') = \pi(a)$ by Lemma 4.2.11 and the parity of P. Suppose that there exists a vertex $v \in K$ with $v < b$. Then, there exists a vertex $u \in L$ such that $u < b$, for otherwise $G_{<b}$ is disconnected (v and a are in different components). Since by Claim 4.3.2, $\pi(u) \neq \pi(b)$ and u has no neighbor with color $\pi(b)$, we have $\pi(u) < \pi(b)$. But then, when the greedy algorithm visits b, color $\pi(u)$ is available for b (because $\pi(u) \neq \pi(a) = \pi(b')$ and u is complete to R), a contradiction. This proves Claim 4.3.3.

By Claim 4.3.3, if $\pi(a) = 1$ or $\pi(a) = 2$, then $\pi(b) = 3 - \pi(a)$ by the parity of P. Since $\mathcal{O}$ is connected, in fact for every vertex v in $V(G) \setminus V(P)$, $v > b$. So, when the greedy algorithm visits $G \setminus P$, the first vertex receives color 1 or 2, and it gives exactly the same colors as the greedy algorithm starting with color 1 or 2 applied to $(G \setminus P, \mathcal{O} \setminus P)$. Hence, by Lemma 4.2.1, we see that $\mathcal{O}$ is a good order for G, a contradiction. Hence, $\pi(a) \geq 3$.

This implies that a has degree at least 2 in $G_{\leq a}$, so a has an in-neighbor u in L. So, if $a' < a$, then $G_{<a}$ is diconnected (a' and u are in different components). Hence, $a < a'$. So there exists a vertex in L with color 1 (to ensure that a has color at least 3). Since $a < a'$, we know by Lemma 4.2.13 that P is well ordered. We therefore have $\pi(b') = \pi(a) \geq 3$ (if $b' = a$) or $\pi(b') = 2$ by the parity of P. This implies $\pi(b) = 1$, contradicting Claim 4.3.2. □

A graph H is an *even birdcage* in G if:

- $V(H) = \cup_{i=1}^{k} V(P_i) \cup C_a \cup C_b$ for some $k \geq 3$.
- $\forall i \in \{1, \ldots, k\}$, P_i is a flat path in G of even length ≥ 2 with two ends a_i, b_i (all a_i's and b_i's are distinct).
- $S_a = \{a_1, \ldots, a_k\}$ and $S_b = \{b_1, \ldots, b_k\}$ are two cliques.
- $K_a = C_a \cup S_a$ and $K_b = C_b \cup S_b$ are two cliques (C_a and C_b may be empty).
- If $C_a \neq \emptyset$, then S_a is a clique cutset of G.
- If $C_b \neq \emptyset$, then S_b is a clique cutset of G.

- These are the only edges in H.
- No vertex in $\cup_{i=1}^{k} V(P_i)$ has a neighbor in $G \setminus H$.

Lemma 4.3.4. *G does not contain an even birdcage.*

Proof. Suppose G contains an even birdcage H, with the notation as in the definition of an even birdcage. Up to symmetry, we suppose $b_k = \max(S_a \cup S_b)$.

If $C_b \neq \emptyset$, let C_1 be the connected component of $G \setminus S_b$ that contains S_a and let C_2 be the connected component of $G \setminus S_b$ that contains C_b. Since G is connected by Lemma 4.2.5, $C_1 \neq C_2$.

If $C_b = \emptyset$, then $G \setminus S_b$ is connected (because G is connected). We then set $C_1 = V(G) \setminus S_b$ and $C_2 = \emptyset$.

Claim 4.3.5. $\max(C_1) < b_k$.

Set $v = \max(C_1)$. If $v \in V(P_i)$ for some $i \in \{1, \dots, k\}$, then $v < b_k$ follows from Lemma 4.2.12. Otherwise, $G_{>\max(S_a)}$ is disconnected (v and b_k are in different components), a contradiction to Lemma 4.2.5. This proves Claim 4.3.5.

For all $i \in \{1, \dots, k\}$, let a_i', b_i' be the vertices in P_i adjacent to a_i, b_i respectively. By Lemma 4.2.14 applied to S_b, we consider the following two cases:

Case 1: For all $i \in \{1, \dots, k\}$, $\pi(b_i') = 1$.

Then, by the parity of P and Lemma 4.2.11, we also have $\pi(a_i') = 1$.

Claim 4.3.6. $\forall i \in \{1, \dots, k\}$, $b_i' < b_i$.

Suppose for some $i \in \{1, \dots, k\}$, $b_i' > b_i$. Then, when the greedy algorithm visits b_i, there must be an in-neighbor of b_i with color 1 (because $\pi(b_i) \neq 1$). This vertex is an in-neighbor of b_k with color 1, a contradiction to Lemma 4.2.14(1). This proves Claim 4.3.6.

Claim 4.3.7. *If $C_b \neq \emptyset$, then $b_k < \min(C_2)$.*

Set $v = \min(C_2)$. We know that $v_1 \in C_1$ by Claim 4.3.6 and Lemma 4.2.8 (applied to S_b). So, $v \in C_b$ since $\mathcal{O}$ is a connected order. Also, since no vertex in S_b has color 1, v receives color 1. Hence, by Lemma 4.2.14(1), $b_k < v$. This proves Claim 4.3.7.

Claim 4.3.8. *C_a contains a vertex x of color 1 (in particular, $C_a \neq \emptyset$).*

By Lemma 4.2.13, we may assume up to symmetry that P_1 is well ordered. Since $\pi(a_1) \neq 1$ and since color 1 does not appear in S_a, a_1 must have an in-neighbor in C_a with color 1. This proves Claim 4.3.8.

Now if $C_b \neq \emptyset$, then $v = \min(C_b)$ receives color 1 and $b_1, \dots, b_k$ are colored with colors $2, \dots, k+1$. This contradicts Lemma 4.2.7. If $C_b = \emptyset$ then $b_k = v_n$ and

$\pi(v_n) = k+1 \leq \chi(G)$ (since $S_a \cup \{x\}$ is a clique of size $(k+1)$ in G), contradicting Lemma 4.2.2.

Case 2: $\pi(b'_i) = 2\ \forall i \in \{1, \ldots, k\}$

Then, by the parity of P and Lemma 4.2.11, we also have $\pi(a'_i) = 2$. By Lemma 4.2.13, up to symmetry, we may assume that P_1 is well ordered. Suppose that $a_1 > a'_1 \geq b'_1 > b_1$ (the case $a_1 < a'_1 \leq b'_1 < b_1$ is similar). Since $\pi(b'_1) = 2$, we have $\pi(b_1) = 1$. Again, by Lemma 4.2.13 and up to symmetry, we may assume that P_2 is well ordered. Since $\pi(b_2) \neq 1$, we must have $b_2 > b'_2 \geq a'_2 > a_2$ and $\pi(a_2) = 1$.

Now if P_3 is also well ordered, we must have $\pi(a_3) = 1$ or $\pi(b_3) = 1$, a contradiction. It follows by Lemma 4.2.13 that the source of G is an internal vertex of P_3. By the parity of P, it follows that $\min(a_3, b_3)$ receives color 1, a contradiction. □

A graph H is an *odd birdcage* in G if:

- $V(H) = C_a \cup C_b \cup (\cup_{i=1}^{k} V(P_i)) \cup (\cup_{i=1}^{m} K_i)$ for some $k \geq 3$, $m \geq 0$.
- P_1 is a path of odd length ≥ 3 with two ends a_1, b_1.
- $\forall i \in \{2, \ldots, k\}$, P_i is a flat path of odd length ≥ 3 with two ends a_i, b_i.
- All a_i's and b_i's are distinct.
- $S_a = \{a_1, \ldots, a_k\}$ and $S_b = \{b_1, \ldots, b_k\}$ are two cliques.
- $K_a = C_a \cup S_a$ and $K_b = C_b \cup S_b$ are two cliques (C_a and C_b might be empty).
- If $C_a \neq \emptyset$, then S_a is a clique cutset of G.
- If $C_b \neq \emptyset$, then S_b is a clique cutset of G.
- For $i \in \{1, \ldots, m\}$, c_i's, d_i's are vertices of P_1 such that:
  - They appear in P_1 in the following order: a_1, c_1, d_1,..., c_m, d_m, b_1.
  - $\forall i \in \{1, \ldots, m\}$, $c_i d_i$ is an edge.
  - $a_1 P_1 c_1$ and $d_m P_1 b_1$ are flat paths of even length ≥ 2.
  - $\forall i \in \{1, \ldots, m-1\}$, $d_i P_1 c_{i+1}$ is a flat path of odd length ≥ 3.
  - $\forall i \in \{1, \ldots, m\}$, K_i is a non-empty clique complete to $\{c_i, d_i\}$.
  - $\forall i \in \{1, \ldots, m\}$, $\{c_i, d_i\}$ is a cutset of G.
- These are the only edges in H.

- No vertex in $\cup_{i=1}^{k} V(P_i)$ has a neighbor in $G \setminus H$.

Lemma 4.3.9. *G does not contain an odd birdcage.*

Proof. Suppose G contains an odd birdcage H as in the definition of an odd birdcage. For $i \in \{1, \ldots, k\}$, let a'_i, b'_i be the neighbors of a_i, b_i in P_i respectively. For $j \in \{1, \ldots, m\}$, let c'_j, d'_j be the neighbors of c_j, d_j in $P_1 \setminus \{c_j, d_j\}$ respectively. Let $x = \max(\cup_{i=1}^{m} V(P_i))$. By Lemma 4.2.12, we may assume that $x = d_i$ for some $i \in \{1, \ldots, m\}$ or $x = b_j$ for some $j \in \{1, \ldots, k\}$.

Case 1: $x = d_i$ for some $i \in \{1, \ldots, m\}$.

By Lemma 4.2.15 applied to the cutset $\{c_i, d_i\}$, there exist some vertices $u \in K_i$, $p, q \notin K_i$ such that u, p, q is a triangle and $\pi(c'_i) = \pi(d'_i) = 2$, $c_i < c'_i$, $u < c_i$ and $u < d_i$. Then $m = i = 1$ for otherwise G contains F_{12}. Let C be the component of $G \setminus \{c_1, d_1\}$ that contains K_1. By Lemma 4.2.8, the source of G is in C. Hence, by Lemma 4.2.13, all flat paths in H are well ordered. In particular, $c'_1 P_1 a_1$ is a directed odd path from c'_1 to a_1. Since $\mathcal{O}$ is connected, $c'_1 P_1 a_1$ contains the first vertices of $\cup_{i=1}^{m} V(P_i)$, so that $\pi(a_1) = 1$. Also, $b_1 P_1 d'_1$ is a directed odd path from b_1 to d'_1, so $\pi(b_1) = 1$ because $\pi(d'_1) = 2$.

Let $b_l = \min(b_1, \ldots, b_k)$. Note that if $C_b \neq \emptyset$, $b_l < \min(C_b)$ for otherwise $G_{<b_l}$ is disconnected (c_1 and $\min(C_b)$ are in different components). So, by the connectivity of $\mathcal{O}$, b_l has an in-neighbor in P_l. Since $b_1 P_1 d'_1$ is a directed odd path from b_1 to d'_1, $b_l \neq b_1$. Also, P_l is well ordered, and since $\pi(a_l) \neq 1$ (because of a_1), we have $\pi(a'_l) = 1$. By the parity of P_l, it follows that $\pi(b_l) = 1$, a contradiction since $\pi(b_1) = 1$.

Case 2: $x = b_j$ for some j.

By Lemma 4.2.14, $\pi(b'_i) = 1 \; \forall i \in \{1, \ldots, k\}$ or $\pi(b'_i) = 2 \; \forall i \in \{1, \ldots, k\}$. Suppose first that $\pi(b'_i) = 1 \; \forall i \in \{1, \ldots, k\}$.

Then for every $i \in \{2, \ldots, k\}$, $\pi(a'_i) = 2$ by the parity of the flat paths. Also, for every $i \in \{2, \ldots, k\}$, $b'_i < b_i$ since otherwise b_i has color ≥ 2 and there must exist in K_b some in-neighbor of b_i having color 1, contradicting Lemma 4.2.14. By Lemma 4.2.13, we may assume that P_3 is well ordered. Since $\pi(a'_3) = 2$, we have $\pi(a_3) = 1$. So, $\pi(a_2) \neq 1$, and since $\pi(a'_2) = 2$, we have $a'_2 < a_2$. So, P_2 is not well ordered. By Lemma 4.2.13, the source of G, v_1, is an internal vertex of P_2 and $v_1 P_2 a_2$ and $v_1 P_2 b_2$ are both well ordered. If $a_2 < b_2$, then $\pi(a_2) = 1$, a contradiction. Hence, $b_2 < a_2$ and $\pi(b_2) = 2$. The vertex v that comes just after b_2 in $\mathcal{O}$ cannot be a_2, because then, again, we would have $\pi(a_2) = 1$. Hence, v is in K_b and receives color 1, a contradiction to Lemma 4.2.14.

Suppose now that $\pi(b'_i) = 2 \; \forall i \in \{1, \ldots, k\}$. Then for every $i \in \{2, \ldots, k\}$, $\pi(a'_i) = 1$ by the parity of the flat paths.

Claim 4.3.10. *If $C_b \neq \emptyset$, then for every vertex $v \in C_b$, $v > b_j$.*

Set $v = \min(C_b)$ and suppose $v < b_j$.

If $\pi(v) = 1$ then $v < b_i$ for every $i \in \{1, \ldots, k\}$ for otherwise, $\min(S_b)$ would receive color 1, a contradiction. Hence, by Lemma 4.2.8, the source of G is in the component of $G \setminus S_b$ containing v. Also, by Lemma 4.2.14, no in-neighbor of b_j in K_b has color 2. So, $\min(S_b)$ has no colored neighbor in $\{b'_1, \ldots, b'_k\}$ when the greedy algorithm visits it, so it receives color 2, a contradiction.

If $\pi(v) \geq 3$, then there exists some vertex $q \notin C_b$ adjacent to v having color 2. If for some i, $b_i < v$, then $G_{<v}$ is not connected (q and b_i are in different components), a contradiction. Then $v < \min(S_b)$, and because $\pi(v) \neq 1$ there exists a vertex $p \notin C_b$ adjacent to v with color 1. We have $pq \in E(G)$ because G is claw-free. Therefore G contains F_{12}, a contradiction. This proves Claim 4.3.10.

Claim 4.3.11. $\pi(d_m) = 1$.

Assume that $\pi(d_m) \neq 1$. We have $\pi(d'_m) = 2$ by the parity of flat path $b'_1 P_1 d'_m$. If $d_m < d'_m$ then $\pi(d_m) = 1$, a contradiction. Therefore, $d'_m < d_m$. If $\pi(b_1) > 1$, then $b'_1 < b_1$ and by Lemma 4.2.13, $v_1 \in b'_1 P_1 d'_m$. So, if $b_1 < d_m$, $\pi(b_1) = 1$, a contradiction, and otherwise $\pi(d_m) = 1$, a contradiction again. Hence, $\pi(b_1) = 1$. Therefore for every $i \neq 1$, $b'_i < b_i$ for otherwise, b'_i receives color 1. Now, consider the cutset $S = \{d_m, b_2, \ldots, b_k\}$. Let C be the component of $G \setminus S$ that contains c_m and D the component that contains b_1. Note that d_m must have an in-neighbor in C, for otherwise it would receive color 1. So, we have $\min(C) < \min(S)$. Also, $\min(D) < \min(S)$ because every vertex in S has an in-neighbor in D (clear for d_m, for the other ones, it follows from the fact that they have an in-neighbor with color 1, namely b_1). This contradicts Lemma 4.2.8. This proves Claim 4.3.11.

Claim 4.3.12. $\pi(c'_m) = 1$.

Suppose $\pi(c'_m) = 2$. Then $c'_m < c_m$ and $d_m < c_m$, so there exists an outneighbor v of c_m in K_m (otherwise $c_m = v_n$, contradicting the maximality of b_j). Therefore $G_{>c_m}$ is disconnected (v and b_j are in different components), contradicting Lemma 4.2.5. This proves Claim 4.3.12.

Similarly, we can prove $\pi(d_i) = \pi(c'_i) = 1$ for all $i \in \{1, \ldots, m\}$, and therefore $\pi(a'_1) = 1$.

We must have a vertex of color 1 in C_a, otherwise $a'_i < a_i$ for every $i \in \{1, \ldots, k\}$. Also, for some $1 \leq i' < i'' \leq k$, we have $b'_{i'} < b_{i'}$ and $b'_{i''} < b_{i''}$ because color 1 appears at most one time in S_b. Hence, G has two sources, a contradiction. So, we have this vertex with color 1 in C_a, and in particular, $C_a \neq \emptyset$, so that $\chi(G) \geq k + 1$.

If $C_b \neq \emptyset$ then let C' be the component of $G \setminus S_b$ that contains C_b. By Claim 4.3.10, we have $\max(S) < \min(C')$. Colors 1 and $3, \ldots, (k+1)$ are used in S_b and color 2 is used for $v = \min(C_b)$. This contradicts Lemma 4.2.7.

If $C_b = \emptyset$, then $b_j = v_n$, but $\pi(b_j) = k+1 \leq \chi(G)$, contradicting Lemma 4.2.2. □

A graph H is a *flower* in G if:

- $V(H) = C_a \cup C_b \cup C_c \cup C_d \cup_{i=1}^{k} V(P_i) \cup_{i=1}^{m} V(Q_i) \cup V(P_{ac}) \cup V(P_{bd})$ for some $k, m \geq 2$.
- For $i \in \{1, \ldots, k\}$, P_i is a flat path of odd length ≥ 3 with two ends a_i and b_i.
- For $i \in \{1, \ldots, m\}$, Q_i is a flat path of odd length ≥ 3 with two ends c_i and d_i.
- P_{ac} is a flat path of even length ≥ 2 with two ends a_0 and c_0.
- P_{bd} is a flat path of even length ≥ 2 with two ends b_0 and d_0.
- All a_i's, b_i's, c_i's, d_i's are distinct.
- $K_a = C_a \cup \{a_0, \ldots, a_k\}$ and $K_b = C_b \cup \{b_0, \ldots, b_k\}$ are cliques.
- $K_c = C_c \cup \{c_0, \ldots, c_m\}$ and $K_d = C_d \cup \{d_0, \ldots, d_m\}$ are cliques.
- If $C_a \neq \emptyset$, $\{a_0, \ldots, a_k\}$ is a clique cutset of G.
- If $C_b \neq \emptyset$, $\{b_0, \ldots, b_k\}$ is a clique cutset of G.
- If $C_c \neq \emptyset$, $\{c_0, \ldots, c_k\}$ is a clique cutset of G.
- If $C_d \neq \emptyset$, $\{d_0, \ldots, d_k\}$ is a clique cutset of G.
- These are the only edges in H.
- No vertex in $G \setminus H$ has a neighbor in $H \setminus (C_a \cup C_b \cup C_c \cup C_d)$.

Lemma 4.3.13. *G does not contain a flower.*

Proof. Suppose G contains a flower H as in the definition of a flower. W.l.o.g, let $b_j = \max(a_0, \ldots, a_k, b_0, \ldots, b_k, c_0, \ldots, c_m, d_0, \ldots, d_m)$. Let a'_i, b'_i, c'_i, d'_i be the unique vertices of degree 2 adjacent to a_i, b_i, c_i, d_i, respectively. By Lemma 4.2.12, $b'_j < b_j$. Applying Lemma 4.2.14, there are two cases:

Case 1: $\pi(b'_i) = 1 \ \forall i \in \{0, \ldots, k\}$.

We omit the proof in this case, because it is similar to the case $\pi(b'_i) = 1$ in Lemma 4.3.9 (but here, we have to consider only two flat paths P_1 and P_2).

Case 2: $\pi(b'_i) = 2\ \forall i \in \{0, \ldots, k\}$: Then for every $i \in \{1, \ldots, k\}$, $\pi(a'_i) = 1$ by the parity of the flat paths.

Claim 4.3.14. *If $C_b \neq \emptyset$, then for every vertex $v \in C_b$, $v > b_j$.*

We omit the proof, because it is similar to the proof Claim 4.3.10 in the proof of Lemma 4.3.9. This proves Claim 4.3.14.

Claim 4.3.15. $\pi(d_0) = 1$.

We omit the proof, because it is similar to the proof of Claim 4.3.11 in the proof of Lemma 4.3.9. This proves Claim 4.3.15.

Let $b_l = \min(b_0, \ldots, b_k)$. It is clear that $\pi(b_l) = 1$ and $b'_i < b_i$ for every $i \in \{0, \ldots, k\} \setminus \{l\}$.

Claim 4.3.16. *For $i \in \{1, \ldots, m\}$, $\pi(d'_i) = 1$.*

W.l.o.g suppose that for some $1 \leq t \leq m$, $\{d'_1, \ldots, d'_t\}$ is the subset of vertices of $\{d'_1, \ldots, d'_m\}$ having color 2. Let $S = \{d_1, \ldots, d_t\}$. We have $d'_i < d_i$ for $i \in \{1, \ldots, t\}$ (otherwise $\pi(d'_i) = 1$). We also have $d_0 < d_i$ for $i \in \{1, \ldots, t\}$ since d_0 is the only vertex of color 1 in K_d. W.l.o.g, assume that $d_t = \max(S)$. There does not exist a vertex u in K_d such that $u < d_t$ and $\pi(u) = 2$ by applying Lemma 4.2.14 for S. Therefore $\min(d_{t+1}, \ldots, d_m) > \max(S)$, otherwise one of them would receive color 2.

If $b_0 = b_l$, applying Lemma 4.2.8 for the cutset $(S \cup \{b_1, \ldots, b_k\}) \cap V(G_{\leq \max(S)})$ of $G_{\leq \max(S)}$, there are sources in both sides of this cutset, a contradiction.

If $b_0 \neq b_l$, applying Lemma 4.2.8 for the cutset $(S \cup \{b_0\}) \cap V(G_{\leq \max(S)})$ of $G_{\leq \max(S)}$, there are sources in both sides of this cutset, a contradiction.

This proves Claim 4.3.16.

By Claim 4.3.16 and the parity of all the flat paths Q_i, $\pi(c'_i) = 2$ for every $i \in \{1, \ldots, m\}$.

Claim 4.3.17. $\pi(c_0) \geq 2$.

Suppose $\pi(c_0) = 1$, then we have $c_0 < c_i$ for $i \in \{1, \ldots, m\}$ since c_0 is the only vertex of color 1 in K_c. We also have $c'_i < c_i$ for $i \in \{1, \ldots, m\}$ since otherwise some vertex in $\{c'_1, \ldots, c'_m\}$ would have color 1.

If $b_0 = b_l$, applying Lemma 4.2.8 for the cutset $\{c_1, \ldots, c_m, b_1, \ldots, b_k\}$ of G, there are sources in both sides of this cutset, a contradiction.

If $b_0 \neq b_l$, applying Lemma 4.2.8 for the cutset $\{c_1, \ldots, c_m, b_0\}$ of G, there are sources in both sides of this cutset, a contradiction. This proves Claim 4.3.17.

Claim 4.3.18. $\pi(c'_0) = 1$.

Suppose $\pi(c'_0) = 2$, then $c'_0 < c_0$. There must exist a vertex v of color 1 in K_c such that $v < c_0$.

If $b_0 = b_l$, applying Lemma 4.2.8 for the cutset $\{c_0, b_1, \ldots, b_k\}$ of G, there are sources in both sides of this cutset, a contradiction.

If $b_0 \neq b_l$, applying Lemma 4.2.8 for the cutset $\{c_0, b_0\}$ of G, there are sources in both sides of this cutset, a contradiction. This proves Claim 4.3.18.

By Claim 4.3.18 and the parity of P_{ac}, $\pi(a'_0) = 1$. We must have a vertex of color 1 in C_a, otherwise $a'_i < a_i$ for every $i \in \{1, \ldots, k\}$ and we have at least two sources in G. Therefore, if $C_b \neq \emptyset$ then $b_0, \ldots, b_k$ receive colors 1 and $3, \ldots, k+2$ and $\min(C_b)$ receives color 2, a contradiction to Lemma 4.2.7. If $C_b = \emptyset$ then $b_j = v_n$, but $\pi(b_j) = k + 2 \leq \chi(G)$ (we have a clique of size $(k + 2)$ in K_a), contradicting Lemma 4.2.2. □

A graph H is a *sun* in G if:

- $V(H) = V(I) \cup_{i=0}^{k} K_i$ for some $k \geq 0$.
- I is a hole.
- For $i \in \{0, \ldots, k\}$, a_i's, b_i's are distinct vertices of I such that:
  - They appear in the following clock-wise order: $a_0, b_0, \ldots, a_k, b_k$.
  - For $\forall i \in \{0, \ldots, k\}$, a_i is adjacent to b_i.
  - For $\forall i \in \{0, \ldots, k\}$, the path in I from b_i to a_{i+1} is a flat path of length ≥ 2 (the subscript is taken modulo $(k + 1)$).
- For $i \in \{0, \ldots, k\}$, K_i is a non-empty clique complete to $\{a_i, b_i\}$.
- For $i \in \{0, \ldots, k\}$, $\{a_i, b_i\}$ is a cutset of G.
- These are the only edges in H.
- No vertex in $G \setminus H$ has a neighbor in I.

Lemma 4.3.19. *G does not contain a sun.*

Proof. Suppose G contains a sun H as in the definition. For $i \in \{0, \ldots, k\}$, let a'_i, b'_i be the vertices of degree 2 adjacent to a_i, b_i in I. W.l.o.g, suppose $b_k = \max(a_0, b_0, \ldots, a_k, b_k)$. Let C_1, C_2 be two connected components of $G \setminus \{a_k, b_k\}$ such that C_2 contains K_k.

By Lemma 4.2.12, $b'_k < b_k$. Applying Lemma 4.2.15 for the cutset $\{a_k, b_k\}$, we have $\pi(a'_k) = \pi(b'_k) = 2$ and there exist some vertex $u \in K_k$, $p, q \in C_2 \setminus K_k$ such that upq is a triangle, $a_k < a'_k$, $u < a_k$ and $u < b_k$. It is clear that $v_1 \in C_2$. If I is an odd hole, then G contains F_{11}, a contradiction. So, the length of I is even. If there exist some a_i, b_i, for $i \neq k$ such that the path $b_k I a_i$ and $b_i I a_k$ (taken in clock-wise order) are of odd length, then G contains F_{12}, a contradiction. Hence, the flat path $b_{k-1} I a_k$ and $b_k I a_0$ are of even length, and each flat path $b_i I a_{i+1}$ is of odd length, for $\forall i \in \{0, \ldots, k-2\}$.

Claim 4.3.20. $\pi(b_{k-1}) = \pi(a'_{k-1}) = 1$.

Since we have $a_k < a'_k$, the flat path $a_k I b_{k-1}$ (in counter-clockwise order) is a directed path in D_G from a_k to b_{k-1}. We have that $\min(K_{k-1} \cup a_{k-1}) > b_{k-1}$, otherwise if there exists a vertex $v \in K_{k-1} \cup a_{k-1}$ such that $v < b_{k-1}$ then $G_{<b_{k-1}}$ is not connected (b'_{k-1} and v are in different connected components). Therefore, $\pi(b_{k-1}) = 1$ since the flat path $a_k I b_{k-1}$ is of even length. We have $a_{k-1} < a'_{k-1}$ since otherwise $G_{<a_{k-1}}$ is not connected (b_{k-1} and a'_{k-1} are in different connected components). Since $\pi(a_{k-1}) \geq 2$, $\pi(a'_{k-1}) = 1$. This proves Claim 4.3.20.

By the same argument as in Claim 4.3.20, we can prove that: $\pi(b_i) = \pi(a'_i) = 1$ for every $i \in \{0, \ldots, k-1\}$. And by Lemma 4.2.11 and the parity of the flat path $a_0 I b_k$, we have $\pi(b'_k) = 1$, contradicting to the fact that $\pi(b'_k) = 2$ which we mentioned previously by Lemma 4.2.15. □

4.4 Proof of Theorem 4.1.3

For our proof, we need results from [31]. A graph G is a *parity graph* if for every pair $u, v \in V(G)$, all induced paths from u to v have the same parity. A graph is *distance-hereditary* if for every pair $u, v \in V(G)$, all induced paths from u to v have the same length. Clearly, every distance-hereditary graph is a parity graph. A graph is *chordal* if it contains no hole.

Theorem 4.4.1 (see [3])**.** *Every gem-free chordal graph is a distance-hereditary graph and therefore a parity graph.*

Theorem 4.4.2 (see [31])**.** *Every fish-free parity graph is good.*

Throughout the rest of this section, let G be a minimally bad claw-free graph that is not an obstruction. Our goal is to prove that this implies a contradiction, thus proving Theorem 4.1.3.

Lemma 4.4.3. *Let $H = v_0 \dots v_k$ be a hole in G and $u \in V(G) \setminus V(H)$ has some neighbor in H. Then u has two or three neighbors in H and they induce a path (of length 1 or 2), or $u \cup V(H)$ induces a 4-wheel.*

Proof. If u is adjacent to some vertex v_i in H, then u must be adjacent also to v_{i-1} or v_{i+1}, otherwise $\{v_i, v_{i-1}, u, v_{i+1}\}$ induces a claw. Suppose $G[N(u) \cap H]$ induces at least two components, where $\{v_i, \dots, v_j\}$ and $\{v_k, \dots, v_m\}$ are its two consecutive components in H (in clock-wise order, i, j, k, m are distinct) then $\{u, v_{j-1}, v_j, \dots, v_k, v_{k+1}\}$ induces F_2, a contradiction. Then $G[N(u) \cap H]$ induces only one component.

- If $|H| \geq 5$: If u has at least four neighbors on H then u and its four consecutive neighbors in H induce a gem (a special case of F_2), a contradiction. Then u has two or three neighbors in H and they induce a path.
- If $|H| = 4$: u can have two or three neighbors in H and they induce a path or u is complete to H and $u \cup V(H)$ induces a 4-wheel.

□

Lemma 4.4.4. *Let $H = v_0 \dots v_k$ be a hole in G. For $i \in \{0, \dots, k\}$, let $S_i = \{u | N(u) \cap H = \{v_{i-1}, v_i, v_{i+1}\}\}$ and $R_i = \{u | N(u) \cap H = \{v_i, v_{i+1}\}\}$. Then for any $i \in \{0, \dots, k\}$:*

1. *S_i is a clique.*
2. *R_i is a clique.*
3. *S_i is complete to S_{i+1} and anticomplete to S_j for any $j \notin \{i-1, i, i+1\}$.*
4. *R_i is anticomplete to R_j for any $j \neq i$.*
5. *S_i is complete to R_{i-1} and R_i.*
6. *If $S_i \neq \emptyset$ then $R_j = \emptyset$ for any $j \notin \{i-1, i\}$.*
7. *If $R_i \neq \emptyset$ then $R_j = \emptyset$ for any $j \in \{i-2, i-1, i+1, i+2\}$.*

Proof. We prove each statement following the index.

1. Let $a, b \in S_i$. If a is not adjacent to b then $\{v_{i+1}, a, b, v_{i+2}\}$ induces a claw, a contradiction.
2. Let $a, b \in R_i$. If a is not adjacent to b then $\{v_{i+1}, a, b, v_{i+2}\}$ induces a claw, a contradiction.

3. Let $a \in S_i$, $b \in S_{i+1}$ and $c \in S_j$. If a is not adjacent to b then $\{v_i, v_{i-1}, a, v_{i+1}, b\}$ induces a gem, a contradiction. If a is adjacent to c then $\{a, c, v_i, v_{i+1}, \ldots, v_{j-1}, v_j\}$ induces F_3 or a gem, a contradiction.

4. Let $a \in R_i$ and $b \in R_j$. If a is adjacent to b then $\{a, b, v_i, v_{i+1}, \ldots, v_j, v_{j+1}\}$ contains F_3 or a gem, a contradiction.

5. Let $a \in S_i$ and $b \in R_i$. If a is not adjacent to b then $\{v_i, v_{i-1}, a, v_{i+1}, b\}$ induces a gem, a contradiction. S_i is also complete to R_{i-1} by symmetry.

6. Let $a \in S_i$ and $b \in R_j$. If a is adjacent to b then $\{a, b, v_i, v_{i+1}, \ldots, v_j, v_{j+1}\}$ contains F_3 or a gem, a contradiction. If a is not adjacent to b then $\{a, b\} \cup V(H)$ contains F_1, F_5, F_6 or a fish (a special case of F_{11}). So, $R_j = \emptyset$.

7. Let $a \in R_i$, $b \in R_{i+1}$ and $c \in R_{i+2}$. If a is not adjacent to b, then $\{a, b\} \cup V(H)$ induces F_2, a contradiction. If a is adjacent to b, then $\{v_{i+1}, v_i, a, b, v_{i+2}\}$ induces a gem, a contradiction. So, $R_{i+1} = \emptyset$. If a is not adjacent to c, then $\{a, c\} \cup V(H)$ induces F_3. If a is adjacent to c, then $\{a, c, v_i, v_{i+1}, v_{i+2}, v_{i+3}\}$ induces either F_3 or F_4, a contradiction. So, $R_{i+2} = \emptyset$. The proof for R_{i-2} and R_{i-1} is similar.

□

Lemma 4.4.5. *G does not contain a 4-wheel.*

Proof. Suppose G contains a 4-wheel consisting of a hole $H = v_0v_1v_2v_3$ and a vertex x complete to that hole. For $i \in \{0, \ldots, 3\}$, let $S_i = \{u | N(u) \cap H = \{v_{i-1}, v_i, v_{i+1}\}\}$, $R_i = \{u | N(u) \cap H = \{v_i, v_{i+1}\}\}$ and $T = \{u | N(u) \cap H = \{v_0, v_1, v_2, v_3\}\}$. Note that $x \in T$.

Claim 4.4.6. *For $i \in \{0, \ldots, 3\}$, $R_i = \emptyset$.*

Let $a \in R_i$. If a is not adjacent to x, then $\{v_i, v_{i-1}, x, v_{i+1}, a\}$ induces a gem, a contradiction. If a is adjacent to x, then $\{x, v_{i-2}, v_{i-1}, v_i, a\}$ induces a gem, a contradiction. So, $R_i = \emptyset$. This proves Claim 4.4.6.

Claim 4.4.7. *For $i \in \{0, \ldots, 3\}$, S_i is complete to T.*

Let $a \in S_i$ and $b \in T$. If a is not adjacent to b then $\{v_{i-1}, v_{i-2}, b, v_i, a\}$ induces a gem, a contradiction. This proves Claim 4.4.7.

By Lemma 4.4.4, S_i is a clique complete to S_{i+1} and anticomplete to S_{i+2}. So, $G[\cup_{i=0}^{3} S_i]$ is P_4-free.

Claim 4.4.8. $V(G) = T \cup_{i=0}^{3} S_i \cup V(H)$.

Suppose there exists some vertex $v \in V(G) \setminus (T \cup_{i=0}^{3} S_i \cup V(H))$ and v has some neighbors in $T \cup_{i=0}^{3} S_i \cup V(H)$. If v has a neighbor in $V(H)$, then $v \in T \cup_{i=0}^{3} S_i$, a contradiction. If v has a neighbor u in S_i for some i but no neighbor in $V(H)$, then $\{u, v, v_{i-1}, v_{i+1}\}$ forms a claw, a contradiction. If v has a neighbor u in $V(T)$ but no neighbor in $V(H)$, then $\{u, v_0, v_2, v\}$ forms a claw, a contradiction. This proves Claim 4.4.8.

We have $G[T]$ is P_4-free (otherwise v_0 and some P_4 in $G[T]$ form a gem), $G[\cup_{i=0}^{3} S_i \cup V(H)]$ is P_4-free and T is complete to $\cup_{i=0}^{3} S_i \cup V(H)$, then G is P_4-free and therefore is good by Theorem 4.1.1, a contradiction. □

A *twin wheel* is a graph consisting of a hole H and a vertex x has three consecutive neighbors on H (see Figure 4.2). Note that this definition of a twin wheel was already mentioned in Section 2.2.

Lemma 4.4.9. *G does not contain a twin wheel.*

Proof. Suppose G contains a twin wheel consisting of a hole $H = v_0 \dots v_k$ and a vertex x that has three neighbors on H: v_0, v_1, v_2. We can assume that H is an even hole since otherwise G contains F_1. For $i \in \{0, \dots, k\}$, let $S_i = \{u | N(u) \cap H = \{v_{i-1}, v_i, v_{i+1}\}\}$ and $R_i = \{u | N(u) \cap H = \{v_i, v_{i+1}\}\}$. We consider two cases:

Case 1: There exists some vertex $y \in R_j$ for some j. Then $j \in \{0, 1\}$ by Lemma 4.4.4. W.l.o.g, suppose $j = 1$ then $R_0 = \emptyset$ by Lemma 4.4.4. Also by Lemma 4.4.4, $S_i = \emptyset$ for $i \notin \{1, 2\}$. Now, $S_1 \cup S_2 \cup R_1 \cup V(H)$ forms a cap in G with $L = S_1 \cup \{v_1\}$, $R = S_2 \cup \{v_2\}$, $C = R_1$ and P is the flat path from v_0 to v_3 in H, contradicting Lemma 4.3.1.

Case 2: For every $i \in \{0, \dots, k\}$, $R_i = \emptyset$. By Lemma 4.4.4, each S_i is a clique complete to S_{i+1}. It is clear that $V(G) = \cup_{i=0}^{k} S_i \cup V(H)$. Hence, G is a parity graph. Also, G is fish-free because the fish is an obstruction. Therefore, G is good by Theorem 4.4.2, a contradiction. □

From now on, by Lemmas 4.4.5 and 4.4.9, the neighborhood of any vertex on a hole in G induces an edge.

Lemma 4.4.10. *G does not contain a short prism.*

Proof. Otherwise, G contains F_3 or F_4, a contradiction. □

From now on, we know that that every prism in G is not short, or in another word, its three paths are of length at least 2.

Lemma 4.4.11. *G does not contain an imparity prism.*

Proof. Otherwise, it contains F_7, a contradiction. □

A graph H is a *prism system* if:

- $V(H) = \cup_{i=1}^{k} V(P_i)$ for some $k \geq 3$.
- $\forall i \in \{1, \ldots, k\}$, P_i is a path of length ≥ 2 with two ends a_i, b_i. All P_i's are disjoint.
- $S_a = \{a_1, \ldots, a_k\}$ and $S_b = \{b_1, \ldots, b_k\}$ are two cliques.
- These are the only edges in H.

Note that if $k = 3$ then H is simply a prism. A prism system is *even* (*odd*) if the lengths of all path P_i's are even (odd).

Lemma 4.4.12. *G does not contain an even prism.*

Proof. Suppose G contains an even prism. Then there exists an even prism system in G as in the description, choose such a prism system H with maximum value of k. Let $C_a = \{v \in V(G) \setminus V(H) | N(v) \cap S_a \neq \emptyset\}$ and $C_b = \{v \in V(G) \setminus V(H) | N(v) \cap S_b \neq \emptyset\}$.

Claim 4.4.13. *All paths P_i's are flat.*

If there exists a vertex $v \in V(G) \setminus V(H)$ which has some neighbors $\{a, b\}$ in the interior of some path P_i, then G contains F_2, F_3 or F_8, a contradiction. This proves Claim 4.4.13.

Claim 4.4.14. *C_a is a clique complete to S_a and C_b is a clique complete to S_b.*

Follows directly from Lemma 4.4.4. This proves Claim 4.4.14.

Claim 4.4.15. *We have the followings:*

1. *If $C_a \neq \emptyset$, S_a is a clique cutset of G.*
2. *If $C_b \neq \emptyset$, S_b is a clique cutset of G.*

We prove only the first statement, the second is similar. Suppose that there exists a path P from some vertex in C_a to some vertex in C_b. The length of P is even otherwise G contains F_7, then $H \cup P$ is a bigger even prism system, a contradiction to the choice of H. This proves Claim 4.4.15.

By Claims 4.4.13, 4.4.14 and 4.4.15, $H \cup C_a \cup C_b$ forms an even birdcage, contradicting Lemma 4.3.4.

□

A *bracelet* (see Figure 4.2) has six paths of length ≥ 2: two paths in the sides are of even length; the other four paths are of odd length.

A graph H is a *bracelet system* if:

- $V(H) = \cup_{i=1}^{k} V(P_i) \cup_{i=1}^{m} V(Q_i) \cup V(P_{ac}) \cup V(P_{bd})$ for some $k, m \geq 2$.
- For $i \in \{1, \ldots, k\}$, P_i is a path of odd length ≥ 3 with two ends a_i and b_i.
- For $i \in \{1, \ldots, m\}$, Q_i is a path of odd length ≥ 3 with two ends c_i and d_i.
- P_{ac} is a path of even length ≥ 2 with two ends a_0 and c_0.
- P_{bd} is a path of even length ≥ 2 with two ends b_0 and d_0.
- All path P_i's, Q_i's, P_{ac}, P_{bd} are disjoint.
- $S_a = \{a_0, \ldots, a_k\}$ and $S_b = \{b_0, \ldots, b_k\}$ are cliques.
- $S_c = \{c_0, \ldots, c_m\}$ and $S_d = \{d_0, \ldots, d_m\}$ are cliques.
- These are the only edges in H.

Note that if $k = m = 2$, then H is simply a bracelet.

Lemma 4.4.16. *G does not contain an odd prism.*

Proof. Suppose G contains an odd prism. We consider the following cases:

Case 1: G contains a bracelet. Then there exists a bracelet system in G as in the description, choose such a system H with maximum value of $k + m$. Let $C_a = \{v \in V(G) \setminus V(H) | N(v) \cap S_a \neq \emptyset\}$, $C_b = \{v \in V(G) \setminus V(H) | N(v) \cap S_b \neq \emptyset\}$, $C_c = \{v \in V(G) \setminus V(H) | N(v) \cap S_c \neq \emptyset\}$ and $C_d = \{v \in V(G) \setminus V(H) | N(v) \cap S_d \neq \emptyset\}$.

Claim 4.4.17. *All the paths P_i's, Q_i's, P_{ac}, P_{bd} are flat.*

Suppose there is some vertex $v \in V(G) \setminus V(H)$ has some neighbor in the interior of one of these paths. If v has some neighbor on P_{ac} or P_{bd}, then G contains F_2, F_3 or F_9, a contradiction. If v has some neighbor on some P_i or Q_i, then G contains F_2, F_3, F_9 or F_{10}, a contradiction. This proves Claim 4.4.17.

Claim 4.4.18. *C_a is a clique complete to S_a.*

Follows directly from Lemma 4.4.4. This proves Claim 4.4.18.

We have a similar statement for C_b, C_c and C_d.

Claim 4.4.19. *If $C_a \neq \emptyset$, S_a is a clique cutset in G.*

Otherwise, if there exists some path P from a vertex in C_a to some vertex in C_c or C_d, then G contains an even prism, contradicting Lemma 4.4.12. If there exists some path P from a vertex in C_a to some vertex in C_b, then P is of odd length and therefore $H \cup P$ is a bigger bracelet system, a contradiction to the choice of H. This proves Claim 4.4.19.

We also have similar statement for S_b, S_c and S_d. By Claims 4.4.17, 4.4.18 and 4.4.19, $H \cup C_a \cup C_b \cup C_c \cup C_d$ is a flower in G, contradicting Lemma 4.3.13.

Case 2: G does not contain bracelet. There exists an odd prism system in G as in the description, choose such a prism system H with maximum value of k. Let $C_a = \{v \in V(G) \setminus V(H) | N(v) \cap S_a \neq \emptyset\}$ and $C_b = \{v \in V(G) \setminus V(H) | N(v) \cap S_b \neq \emptyset\}$.

Claim 4.4.20. *Let $v \in V(G) \setminus V(H)$ be a vertex has some neighbor $\{a, b\}$ in the interior of some path P_i (a is closer to a_i than b in P_i). Then two paths $a_i P_i a$ and $b_i P_i b$ are of even length ≥ 2.*

Otherwise G contains F_2, F_3 or F_9, a contradiction. This proves Claim 4.4.20.

Claim 4.4.21. *If $v \in V(G) \setminus V(H)$ has some neighbors in P_i, then all path P_j's are flat for any $j \neq i$.*

Otherwise G contains F_{10}, a contradiction. This proves Claim 4.4.21.

W.l.o.g, suppose that P_1 is the only path among P_i's which might not be flat. For some $m \geq 0$, let $\{c_1, d_1\}, \ldots, \{c_m, d_m\}$ be all the possible positions in P_1 to which a vertex $v \in V(G) \setminus V(H)$ can be adjacent ($c_i d_i$ is an edge; all vertices are listed in order from a_1 to b_1). For $i \in \{1, \ldots, m\}$, let $K_i = \{v \in V(G) \setminus V(H) | N(v) \cap \{c_i, d_i\} \neq \emptyset\}$.

Claim 4.4.22. *C_a is a clique complete to S_a; C_b is a clique complete to S_b and K_i is a clique complete to $\{c_i, d_i\}$ for $i \in \{1, \ldots, m\}$.*

Follows directly from Lemma 4.4.4. This proves Claim 4.4.22.

Claim 4.4.23. *$a_1 P_1 c_1$ and $d_m P_1 b_1$ are flat paths of even length ≥ 2; $d_i P_1 c_{i+1}$ is a flat path of odd length ≥ 3 for $i \in \{1, \ldots, m-1\}$.*

Follows from Claim 4.4.20. This proves Claim 4.4.23.

Claim 4.4.24. *If C_a and $C_b \neq \emptyset$, S_a, S_b, $\{c_i, d_i\}$ are clique cutsets in G for $i \in \{1, \ldots, m\}$.*

Otherwise, if there is a path from some vertex in K_i to some vertex in K_j for some $i \neq j$, then G contains a bracelet, a contradiction. If there is a path from some vertex in C_a to some vertex in K_i for some i, then G contains F_7 or an even prim, a contradiction. If there is a path P from a vertex in C_a to some vertex in C_b, then

$H \cup P$ is a bigger odd prism system, a contradiction to the choice of H. This proves Claim 4.4.24.

By Claims 4.4.21, 4.4.22, 4.4.23 and 4.4.24, $H \cup C_a \cup C_b \cup_{i=1}^{m} K_i$ forms an odd birdcage in G, contradicting Lemma 4.3.9. □

By Lemmas 4.4.10, 4.4.11, 4.4.12 and 4.4.16, G is prism-free.

Lemma 4.4.25. *G does not contain a hole.*

Proof. Suppose G contains a hole I. For some $k \geq 0$, let $\{a_0, b_0\}, \ldots, \{a_k, b_k\}$ be all the possible positions in I to which a vertex $v \in V(G) \setminus I$ can be adjacent ($a_i b_i$ is an edge; all the vertices are listed in clock-wise order). For $i \in \{0, \ldots, k\}$, let $K_i = \{v \in V(G) \setminus I | N(v) \cap \{a_i, b_i\} \neq \emptyset\}$.

Claim 4.4.26. *For $i \in \{0, \ldots, k\}$, K_i is a clique complete to $\{a_i, b_i\}$.*

Follows directly from Lemma 4.4.4. This proves Claim 4.4.26.

Claim 4.4.27. *For $i \in \{0, \ldots, k\}$, $\{a_i, b_i\}$ is a cutset of G.*

Otherwise, there is a path from a vertex in K_i to some vertex in K_j, for some $j \neq i$, so G contains a prism, a contradiction. This proves Claim 4.4.27.

By Claims 4.4.26 and 4.4.27, $I \cup_{i=0}^{k} K_i$ forms a sun in G, contradicting Lemma 4.3.19. □

By Lemmas 4.4.25, G is chordal. And since G is gem-free, G is a parity graph by Theorem 4.4.1. Hence G is a fish-free parity graph and is therefore good by Theorem 4.4.2, a contradiction. This proves that every minimally bad claw-free graph is an obstruction.

To prove Theorem 4.1.3, we are left to prove that every obstruction is a minimally bad claw-free graph. Suppose that it is not true for some graph F in the list of obstructions. Since F is bad (as we already specify a bad order for every obstruction), F must contain a minimally bad claw-free graph F' as an induced subgraph. Since every minimally bad claw-free graph is an obstruction, F' is also an obstruction. However, it is easy to check that there do not exist two obstructions in our list such that one contains the other as an induced subgraph, a contradiction.

4.5 Conclusion

In this chapter, we give the characterization of good claw-free graphs in terms of minimal forbidden induced subgraphs. Note that the arguments in Sections 4.3 and

4.4 can be turned into a polynomial algorithm for recognizing this class, where each structure in Section 4.3 corresponds to a kind of decomposition. A full characterization of good graphs seems hard to achieve, as we observe that the actual structure of minimally bad graphs could be much more complicated. The following question is open:

Open question. Is the chromatic number of every minimally bad graph 3?

We see that this is true for claw-free graphs. The next step would be finding the characterization for good perfect graphs, or some interesting subclasses of perfect graphs.

Conclusion

Throughout the thesis, we have studied the coloring and recognition problem for several graph classes. In this last chapter, let us review all these and discuss some open questions.

In Chapter 2, we focus on ISK4-free graphs. The study of this class is motivated by Scott's question [45] on χ-boundedness of graph classes defined by forbidding all the subdivisions of a fixed graphs and the dichotomy between polynomial and NP-hardness on detecting an induced subdivision [37]. We prove that the chromatic number of ISK4-free graphs is bounded by some small constant (and even smaller in triangle-free case) and propose a polynomial-time recognition algorithm. Recall that $Forb^*(H)$ is a graph class defined by forbidding all subdivision of H as induced subgraphs. There are examples of graph H such that $Forb^*(H)$ is χ-bounded and examples of graph H where $Forb^*(H)$ is not χ-bounded. Similarly, there exist some examples of graph H such that $Forb^*(H)$ can be recognized in polynomial-time and examples of graph H where $Forb^*(H)$ is NP-hard to detect. However, the complete characterization of whether $Forb^*(H)$ is χ-bounded and whether $Forb^*(H)$ is recognizable in polynomial-time is still very far from completion. We do not even know an instance of a subcubic graph H where detecting $Forb^*(H)$ is NP-hard. In other words, we do not know if it is possible to detect $Forb^*(H)$ in polynomial-time for every subcubic graph H.

In Chapter 3, we studied even-hole-free graphs with no star cutset. This was motivated by even-hole-free graphs (a graph class which is closely related to perfect graphs) and the use of decomposition technique in solving some optimization problems. We prove the optimal χ-bounding function for even-hole-free graphs with no star cutset and also show that this class has bounded rank-width, which implies the existence of a polynomial-time coloring algorithm. The complexity of finding a maximum stable set and an optimal coloring for even-hole-free graphs remains open, even though we know the positive answers for a number of its subclasses.

In Chapter 4, connected greedy coloring is considered. This was motivated by the greedy coloring algorithm applied to some order of the vertices. Many researches

have been done for general orders (see [10, 21, 52, 7]): the complexity of computing Grundy number, the characterization of graphs where there exists a bad ordering, the complexity of recognizing (hereditary) well-colored graphs, ... Connected orders have also been studied [31, 2, 4, 7]. We know that it is NP-hard to compute connected Grundy number. Some examples of graphs that are not friendly with connected orders were also given. However, we know very little about the characterization of good graphs with respect to connected orders. In this chapter, we gave the complete characterization of a good claw-free graphs in terms of minimal forbidden induced subgraphs. This implies also a polynomial-time algorithm for recognizing good claw-free graphs. We would like to know the complexity of recognizing and the full characterization of good graphs in general. The next direction should be to study this question for some interesting classes, for example perfect graphs.

Contributions

All the results in this thesis are covered in the following publications:

[I] N.K. Le. Chromatic number of ISK4-free graphs, *Graphs and Combinatorics*, 33(6): 1635-1646, 2017.

[II] N.K. Le. Detecting an induced subdivision of K_4, *arXiv preprint*, submitted. `https://arxiv.org/abs/1703.04637`

[III] N.K. Le. Coloring even-hole-free graphs with no star cutset, *arXiv preprint*, submitted.

[IV] N.K. Le, N. Trotignon. Connected greedy colouring in claw-free graphs, *arXiv preprint*, submitted.

Bibliography

[1] L. Addario-Berry, M. Chudnovsky, F. Havet, B. Reed, and P. Seymour. Bisimplicial vertices in even-hole-free graphs. *Journal of Combinatorial Theory, Series B*, 98(6):1119–1164, 2008.

[2] L. Babel and G. Tinhofer. Hard-to-color graphs for connected sequential colorings. *Discrete Applied Mathematics*, 51(1-2):3–25, 1994.

[3] H. Bandelt and H. M. Mulder. Distance-hereditary graphs. *J. Comb. Theory, Ser. B*, 41(2):182–208, 1986.

[4] F. Benevides, V. Campos, M. Dourado, S. Griffiths, R. Morris, L. Sampaio, and A. Silva. Connected greedy colourings. In *Latin American Symposium on Theoretical Informatics*, pages 433–441. Springer, 2014.

[5] C. Berge. Färbung von graphen, deren sämtliche bzw. deren ungerade kreise starr sind. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 10(114):88, 1961.

[6] J. A. Bondy and U. S. R. Murty. *Graph theory*, volume 244. Springer, 2008.

[7] É. Bonnet, F. Foucaud, E. J. Kim, and F. Sikora. Complexity of grundy coloring and its variants. In *International Computing and Combinatorics Conference*, pages 109–120. Springer, 2015.

[8] K. Cameron, S. Chaplick, and C. T. Hoàng. On the structure of (pan, even hole)-free graphs. *Journal of Graph Theory*, 87(1):108–129, 2018.

[9] J. Chalopin, L. Esperet, Z. Li, and P. Ossona de Mendez. Restricted frame graphs and a conjecture of Scott. *arXiv preprint arXiv:1406.0338*, 2014.

[10] C. A. Christen and S. M. Selkow. Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B*, 27(1):49–59, 1979.

[11] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[12] M. Chudnovsky, C.-H. Liu, O. Schaudt, S. Spirkl, N. Trotignon, and K. Vuskovic. Triangle-free graphs that do not contain an induced subdivision of K_4 are 3-colorable. *arXiv preprint arXiv:1704.08104*, 2017.

[13] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.

[14] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. *Combinatorica*, 30(4):387–417, 2010.

[15] M. Chudnovsky, P. Seymour, and N. Trotignon. Detecting an induced net subdivision. *Journal of Combinatorial Theory, Series B*, 103(5):630–641, 2013.

[16] M. Chudnovsky, N. Trotignon, T. Trunck, and K. Vušković. Coloring perfect graphs with no balanced skew-partitions. *Journal of Combinatorial Theory, Series B*, 115:26–65, 2015.

[17] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs part I: Decomposition theorem. *Journal of Graph Theory*, 39(1):6–49, 2002.

[18] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs part II: Recognition algorithm. *Journal of graph theory*, 40(4):238–266, 2002.

[19] G. Cornuéjols and W. Cunningham. Compositions for perfect graphs. *Discrete Mathematics*, 55(3):245–254, 1985.

[20] M. V. Da Silva and K. Vušković. Decomposition of even-hole-free graphs with star cutsets and 2-joins. *Journal of Combinatorial Theory, Series B*, 103(1):144–183, 2013.

[21] D. de Werra. Heuristics for graph colorings. In *Computational Graph Theory, Computing Supplementum*, volume 7, pages 191–208. Springer, 1990.

[22] R. Diestel. *Graph theory*. Springer Publishing Company, Incorporated, 2017.

[23] E. Diot, M. Radovanović, N. Trotignon, and K. Vušković. The (theta, wheel)-free graphs Part I: only-prism and only-pyramid graphs. *arXiv preprint arXiv:1504.01862*, 2015.

[24] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.

[25] P. Erdős. Graph theory and probability. *Canad. J. Math*, 11(11):34–38, 1959.

[26] M. Farber. On diameters and radii of bridged graphs. *Discrete Mathematics*, 73(3):249–260, 1989.

[27] M. C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(03):423–443, 2000.

[28] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[29] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[30] A. Gyárfás. Problems from the world surrounding perfect graphs. *Applicationes Mathematicae*, 19(3-4):413–441, 1987.

[31] A. Hertz and D. de Werra. Connected sequential colorings. *Discrete mathematics*, 74(1-2):51–59, 1989.

[32] I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on computing*, 10(4):718–720, 1981.

[33] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[34] D. Kobler and U. Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.

[35] J. Kratochvíl and A. Sebo. Coloring precolored perfect graphs. *Journal of Graph Theory*, 25(3):207–215, 1997.

[36] D. Kühn and D. Osthus. Induced subdivisions in $K_{s,s}$-free graphs of large average degree. *Combinatorica*, 24(2):287–304, 2004.

[37] B. Lévêque, D. Y. Lin, F. Maffray, and N. Trotignon. Detecting induced subgraphs. *Discrete Applied Mathematics*, 157(17):3540–3551, 2009.

[38] B. Lévêque, F. Maffray, and N. Trotignon. On graphs with no induced subdivision of K_4. *Journal of Combinatorial Theory, Series B*, 102(4):924–947, 2012.

[39] J. Mycielski. Sur le coloriage des graphes. In *Colloq. Math*, volume 3, page 9, 1955.

[40] S.-I. Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms (TALG)*, 5(1):10, 2008.

[41] S.-I. Oum and P. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

[42] A. Pawlik, J. Kozik, T. Krawczyk, M. Lasoń, P. Micek, W. T. Trotter, and B. Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *Journal of Combinatorial Theory, Series B*, 105:6–10, 2014.

[43] N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[44] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[45] A. D. Scott. Induced trees in graphs of large chromatic number. *Journal of Graph Theory*, 24(4):297–311, 1997.

[46] D. Seinsche. On a property of the class of n-colorable graphs. *Journal of Combinatorial Theory, Series B*, 16:191–196, 1974.

[47] N. Trotignon. Personal communication. 2015.

[48] N. Trotignon and K. Vušković. Combinatorial optimization with 2-joins. *Journal of Combinatorial Theory, Series B*, 102(1):153–185, 2012.

[49] N. Trotignon and K. Vušković. On triangle-free graphs that do not contain a subdivision of the complete graph on four vertices as an induced subgraph. *Journal of Graph Theory*, 2016.

[50] K. Vušković, N. Trotignon, M. Radovanović, H. Müller, N. K. Le, and I. Adler. On rank-width of even-hole-free graphs. *Discrete Mathematics & Theoretical Computer Science*, 19, 2017.

[51] K. Vušković. The world of hereditary graph classes viewed through Truemper configurations. *Surveys in Combinatorics 2013*, 409:265, 2013.

[52] M. Zaker. Results on the grundy chromatic number of graphs. *Discrete mathematics*, 306(23):3166–3173, 2006.