



HAL
open science

On various graph coloring problems

Dimitri Lajou

► **To cite this version:**

Dimitri Lajou. On various graph coloring problems. Data Structures and Algorithms [cs.DS]. Université de Bordeaux, 2021. English. NNT : 2021BORD0339 . tel-03522406

HAL Id: tel-03522406

<https://theses.hal.science/tel-03522406>

Submitted on 12 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE
DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET
D'INFORMATIQUE
Spécialité : INFORMATIQUE

Par **Dimitri LAJOU**

Sur divers problèmes de coloration de graphes
On various graph coloring problems

Sous la direction de : **Hervé HOCQUARD** et **Éric SOPENA**

Soutenue le 10 décembre 2021

Membres du jury :

Mme Colette JOHNEN	Professeure des universités	Université de Bordeaux	Présidente
Mme Aline PARREAU	Chargée de recherche	Université Lyon 1	Examinatrice
M. Alexandre PINLOU	Professeur des universités	Université de Montpellier	Rapporteur
M. Sagnik SEN	Assistant professor	IIT Dharwad	Examineur
M. Mariusz WOŹNIAK	Professor	AGH Kraków	Rapporteur
M. Hervé HOCQUARD	Maître de conférences	Université de Bordeaux	Directeur
M. Éric SOPENA	Professeur des universités	Université de Bordeaux	Directeur

Titre : Sur divers problèmes de coloration de graphes.

Résumé Dans cette thèse, nous étudions des problèmes de coloration de graphe. Nous nous intéressons à deux familles de colorations.

La première consiste à colorer des graphes, appelés graphes signés, modélisant des relations sociales. Ceux-ci disposent de deux types d'arêtes : les arêtes positives pour représenter l'amitié et les arêtes négatives pour l'animosité. Nous pouvons colorer des graphes signés à travers la notion d'homomorphisme : le nombre chromatique d'un graphe signé (G, σ) est alors le nombre minimum de sommets d'un graphe signé (H, π) tel que (G, σ) admet un homomorphisme vers (H, π) . Nous étudions la complexité des homomorphismes de graphes signés quand la cible est fixée et quand l'entrée peut être modifiée, et obtenons des dichotomies P/NP-complet et FPT/W[1]-difficile. Nous obtenons des bornes supérieures sur le nombre chromatique d'un graphe signé quand le graphe a peu de cycles. Enfin, nous étudions les relations entre les homomorphismes de graphes signés et le produit Cartésien des graphes signés.

La deuxième famille de coloration consiste à colorer les arêtes au lieu des sommets en respectant différents critères. Nous étudions quatre types de colorations d'arêtes : la coloration d'arêtes « packing », la coloration d'arêtes injective, la coloration AVD et les 1-2-3-étiquetages. La coloration d'arêtes « packing » est une forme de coloration propre d'arêtes où chaque couleur a ses propres règles de conflits, par exemple, la couleur 1 pourrait obéir aux règles de la coloration propre d'arêtes tandis que la couleur 2 obéirait aux règles de la coloration forte d'arêtes. Nous étudions cette forme de coloration sur les graphes subcubiques en donnant des bornes supérieures sur le nombre de couleurs nécessaires pour colorer ces graphes. Une coloration d'arêtes injective est une coloration d'arêtes telle que pour chaque chemin de longueur 3, les deux arêtes aux extrémités du chemin n'ont pas la même couleur. Nous déterminons la complexité de la coloration d'arêtes injective sur plusieurs classes de graphes. Pour les colorations AVD, c'est-à-dire les colorations propres d'arêtes où les sommets adjacents sont incidents à des ensembles de couleurs différents, nous obtenons des bornes supérieures sur le nombre de couleurs requises pour colorer le graphe quand le degré maximum du graphe est significativement plus grand que son degré moyen maximum, ou quand le graphe est planaire et a un degré maximum supérieur ou égal à 12. Finalement, nous prouvons la 1-2-3 Conjecture multiplicative : pour tout graphe connexe (non réduit à une arête), on peut colorer ses arêtes avec les couleurs 1, 2 et 3 de telle manière que la coloration (de sommets) obtenue en associant à un sommet le produit des couleurs de ses arêtes incidentes est propre.

Mots-clés graphe, coloration, graphe signé, homomorphisme, coloration d'arêtes.

Title: On various graph coloring problems.

Abstract In this thesis, we study some graph coloring problems. We are interested in two families of colorings.

The first one consists in coloring graphs, called signed graphs, modeling social links. These signed graphs dispose of two types of edges: positive edges to represent friendship and negative edges for animosity. Coloring signed graphs is done through the notion of homomorphism: the chromatic number of a signed graph (G, σ) is the smallest order of a signed graph (H, π) to which (G, σ) admits a homomorphism. We study the complexity of homomorphisms of signed graphs when the target graph is fixed and when the input can be modified, giving P/NP-complete dichotomies and FPT/W[1]-hard dichotomies. We also present bounds on the chromatic number of signed graphs when the input graph has few cycles. Finally, we study the relationship between homomorphisms of signed graphs

and the Cartesian product of signed graphs.

The second family of colorings consists in coloring edges, instead of vertices, according to some constraints. We study four kinds of edge-colorings notions: packing edge-colorings, injective edge-colorings, AVD colorings and 1-2-3-labellings. Packing edge-coloring is a form of proper edge-coloring where each color has its own conflict rule, for example, color 1 may behave according to the rules of proper edge-colorings while color 2 behave according to the rules of strong edge-colorings. We study packing edge-coloring on subcubic graphs and provide bounds on the number of colors necessary to color the graphs. An injective edge-coloring is an edge-coloring where for any path of length 3, the two non-internal edges of the path receive different colors. We determine the complexity of injective edge-coloring for some classes of graphs. For AVD colorings, *i.e.* a proper edge-coloring where adjacent vertices are incident with different sets of colors, we obtain bounds on the number of colors required to color the graph when the graph has its maximum degree significantly greater than its maximum average degree and when the graph is planar and has maximum degree at least 12. Finally, we prove the Multiplicative 1-2-3 Conjecture, *i.e.* that every connected graph (which is not just an edge) can be edge-labelled with labels 1, 2 and 3 so that the coloring of G , obtained by associating with each vertex the product of the labels on edges incident with u , is proper.

Keywords graph, coloring, signed graph, homomorphism, edge-coloring.

Remerciements

Cette thèse n'aurait pas été possible sans le soutien de plusieurs personnes.

J'aimerais tout d'abord remercier ma famille. Merci maman pour ton soutien moral, merci papa pour tes nombreux conseils et pour les nombreuses fois où tu es venu me dépanner, merci tonton Francis pour ton naturel et ta gentillesse. Merci aussi à toute ma famille maternelle (oncles, tantes, cousins...) que je ne peux malheureusement pas citer un à un au risque de doubler la taille de cette thèse.

J'aimerais ensuite remercier énormément Hervé et Éric pour le soutien que vous m'avez apporté pendant ces dernières années. Merci de m'avoir laissé beaucoup de liberté dans mon travail et notamment sur le choix des sujets abordés. Vous avez été très patients avec moi et avez relu beaucoup de preuves pas forcément très bien écrites et j'aime croire que je me suis amélioré sur ce point-là grâce à vos nombreuses remarques.

J'aimerais ensuite remercier Mariusz et Alexandre d'avoir rapportés ma thèse. Compte tenu du nombre de pages, cela n'a pas dû être une tâche facile. J'aimerais ensuite remercier aussi les autres membres du jury: Colette, Aline et Sagnik. Ce fut un plaisir pour moi d'avoir pu faire ma soutenance en physique, et je remercie donc tous les membres du jury d'avoir fait le déplacement.

I would also like to thank all of my co-authors without which I would not have been able to produce this thesis. Thanks Borut for your hospitality, I hope that I will be able to work with you in Slovenia once again. Thanks to my Indian co-authors: Dibyayan, Sandip, Harmender and Bodhayan. Thanks also to Jan and Nikola, it was cool to work with you. Merci encore à Hervé et Éric. Deux grands mercis à Florent et Julien, j'ai eu l'occasion de travailler avec vous deux sur différents projets et ce fut toujours un plaisir. Merci aussi à Marthe, Nicolas, Valia, Théo et Jonathan avec qui j'ai travaillé plus ponctuellement.

J'aimerais aussi remercier tous les collègues de l'équipe CombAlgo ainsi que ceux de l'IUT avec qui j'ai eu l'occasion de préparer des cours. Un remerciement tout particulier à tous les autres doctorants (et ceux qui sont devenus docteurs entre-temps) de l'équipe pour la bonne ambiance qui règne au labo.

Enfin, je voudrais tout particulièrement remercier mes collègues du bureau 325 sans qui l'expérience n'aurait pas été si agréable. Merci à Théo pour avoir montré l'exemple et pour ta bonne humeur. Merci à Antonin pour tes innombrables conseils que ce soit sur l'enseignement ou pour naviguer dans les différents dédales administratifs. Merci à Alex pour avoir animé le bureau et pour les innombrables discussions que nous avons eues. Merci à Tobias pour ta prise d'initiative pour nous réunir en dehors du travail sur CS et pour m'avoir invité à participer à ta campagne sur D&D. Au passage merci à Jason, Benoît, Uli et Yumei pour la bonne humeur lors de ces soirées jeu de rôle. Merci aussi aux autres personnes du bureau qui ont défilées au fil des ans : Victor, Ibrahim, Sabri, Rémi, Théo M. ...

Merci aussi à tous ceux que j'ai oublié de mentionner explicitement mais qui ont aussi été une part importante de ma vie lors de ces années de thèse.

Contents

Introduction (en français)	9
Introduction (in english)	13
1 Preliminaries	17
1.1 Graphs: definitions, notation	18
1.2 Colorings and Homomorphisms	30
1.3 NP-completeness and FPT	38
1.4 A little bit of algebra	46
I Signed Graphs	51
2 Introduction to signed graphs	53
2.1 Key definitions	54
2.2 Homomorphisms and coloring of signed graphs	58
2.3 Classes of cycles	63
2.4 Complexity	64
2.5 Similar notions and generalizations	65
3 Complexity of edge-colored and signed graphs modification problems	67
3.1 Preliminaries	72
3.2 Edge-colored modification problems	77
3.3 LIMITED SWITCHINGS (H, π) -COLORING when H has order 2	87
3.4 VERTEX DELETION SIGNED- (H, π) -COLORING	94
3.5 EDGE DELETION SIGNED- (H, π) -COLORING	115
3.6 Conclusion and perspectives	117
4 Coloring signed graphs with small cyclomatic number	119
4.1 Proofs of Theorems 4.8, 4.9 and 4.10	124
4.2 Proof of Theorem 4.11 and 4.12	130
4.3 Proof of Theorem 4.13	133
4.4 Perspectives	135
5 Cartesian product of signed graphs	137
5.1 Definitions and preliminary results	138
5.2 Cartesian products of signed graphs	139
5.3 Chromatic number of Cartesian products of complete signed graphs and upper bounds	149

5.4	Chromatic number of Cartesian products of signed cycles	153
5.5	Perspectives	167
II	Edge-coloring with constraints	169
6	Introduction	171
6.1	Distance edge-coloring	171
6.2	Distinguishing vertices with edge-colors	172
7	Between proper and strong edge-colorings of subcubic graphs	175
7.1	Preliminaries	178
7.2	Proofs of Theorem 7.5.1 and Theorem 7.5.3	179
7.3	Proof of Theorem 7.5.2	180
7.4	Proof of Theorem 7.5.4	184
7.5	Further Work	200
8	Complexity of the injective edge-coloring problem	205
8.1	NP-completeness for cubic graphs	208
8.2	Complexity results for planar graphs	211
8.3	INJECTIVE k -EDGE-COLORING is FPT when parameterized by the treewidth	214
8.4	INJECTIVE k -EDGE-COLORING is NP-complete even for graphs with maximum degree $O(\sqrt{k})$	216
8.5	Conclusion	218
9	Adjacent vertex-distinguishing edge coloring of graphs	221
9.1	Preliminaries	224
9.2	Sketch of the proofs	225
9.3	Proof of the reducibility lemmas	229
9.4	Proof of Lemma 9.20	243
9.5	Discharging procedures	252
9.6	Perspectives	264
10	The Multiplicative 1-2-3 Conjecture	265
10.1	Proof of the product 1-2-3 Conjecture	267
10.2	A conjecture for 2-labellings with restricted product conflicts	280
10.3	On the list variant of p-proper labellings	282
10.4	Improved bounds on ch_{Π}^* for some graph classes	287
10.5	Perspectives	306

Introduction (en français)

Le but de cette thèse est d'étudier certains problèmes de coloration de graphe. Un *graphe* (fini) est un objet mathématique composé d'un nombre fini de sommets et d'un nombre fini d'arêtes reliant des paires de sommets. Les graphes sont un outil d'abstraction utilisé dans plusieurs domaines, on pourra par exemple citer : les réseaux routiers, les réseaux de télécommunications, l'analyse de communautés, les bases de données, l'analyse ADN...

L'une des applications des graphes se trouve en psychologie sociale. Cette dernière étudie la dynamique des relations entre diverses entités. Par exemple, les graphes peuvent être utilisés pour décrire les relations entre les pays au cours de la première guerre mondiale (voir [7]). Pour cela, on peut utiliser un type de graphe avec deux sortes d'arêtes, celles-ci étant soit positives soit négatives, les arêtes positives représentant l'amitié et les arêtes négatives représentant l'animosité. On appelle un tel graphe un *graphe signé* et on le note généralement sous la forme d'un couple (G, σ) . Les graphes signés ont été introduits par Harary en 1953 [92].

Supposons que nous avons trois pays (ou personnes) A , B et C , chacun ayant des relations avec les deux autres. Certains choix pour les relations forment des situations stables alors que d'autres sont instables. Cela se généralise bien entendu avec plus de trois pays (voir [7]). L'étude de ces relations nous permet de déterminer si une configuration est stable ou instable. Les quatre situations possibles entre ces trois pays sont représentées sur la Figure 1.

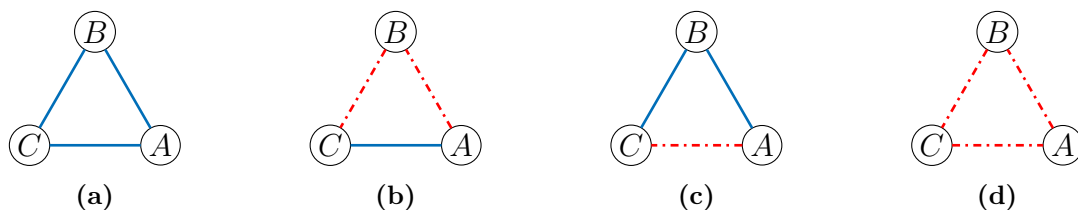


Figure 1: Les quatre relations possibles entre A , B et C . Les arêtes positives sont dessinées avec des traits pleins bleus et les arêtes négatives sont dessinées avec des pointillés rouges.

Analysons cela en détail.

- Dans le premier cas, les trois pays sont alliés (voir Figure 1(a)). Dans ce cas, la situation est stable. Cela ne signifie pas que les relations entre les pays ne peuvent pas se détériorer mais qu'une détérioration des relations proviendrait de facteurs extérieurs et ne saurait être due à des instabilités du graphe. Il y a beaucoup d'exemples de telles situations stables. On peut par exemple évoquer l'Union Européenne où chaque pays membre est allié aux autres.
- Dans le deuxième cas, deux pays, disons A et C , sont alliés contre le troisième (voir Figure 1(b)). Cette situation est elle aussi stable. C'est l'exemple classique du bloc

contre bloc en géopolitique. L’alliance entre le Rohan et le Gondor contre les armées de Sauron dans *Le Seigneur des anneaux : Le Retour du roi* en est un exemple.

- Le troisième cas se déroule quand deux pays, disons A et C , sont ennemis alors que le troisième, B , est allié avec chacun d’entre eux (voir Figure 1(c)). Nous avons ici notre première situation instable : B est forcé de choisir entre son alliance avec A et son alliance avec C . Il est aussi possible que A et C se réconcilient grâce à leur ami commun. Dans tous les cas, la situation est condamnée à changer à cause des relations entre A , B et C . Par exemple, dans *Star Wars, épisode III : La Revanche des Sith*, Anakin a été forcé de choisir entre Palpatine et l’ordre des Jedi.
- Enfin, le quatrième cas se produit quand les trois pays sont ennemis (voir Figure 1(d)). Cette situation est elle aussi instable. En effet, comme dit le dicton : « l’ennemi de mon ennemi est mon ami ». Il est alors très probable que deux des trois pays s’allient contre le troisième. Par exemple, dans *Game of Thrones*, la Garde de Nuit, le Peuple libre et les Marcheurs Blancs étaient en guerre les uns contre les autres jusqu’à ce que les deux premiers s’allient contre le troisième pour survivre.

Ces exemples, présentés avec trois pays, peuvent se généraliser à un nombre arbitraire d’intervenants. On peut aussi vouloir considérer le cas où deux pays n’ont pas de relations (bonnes ou mauvaises). Pour représenter ce cas, on préfère n’avoir aucune arête entre A et B dans le graphe signé.

La notion de « stabilité », formalisée par Harary [92] sous le nom de « *balance* » (équilibre), nous permet d’étudier les graphes signés. Il est intéressant de pouvoir comparer les graphes signés, c’est-à-dire vérifier si deux graphes signés présentent les mêmes structures stables et instables. De là découle la notion d’homomorphisme de graphes signés. Informellement, il y a un homomorphisme entre deux graphes signés (G, σ) et (H, π) si on peut envoyer les cycles stables (respectivement instables) de (G, σ) vers des cycles stables (respectivement instables) de (H, π) . Appliquer des homomorphismes nous permet de catégoriser les sommets, c’est-à-dire regrouper les sommets d’un graphe signé qui se comportent d’une manière similaire.

Les homomorphismes de graphes signés (et de certaines notions dérivées) est l’objet d’étude de la première partie de cette thèse.

La seconde partie de cette thèse est dédiée aux colorations d’arêtes. L’objectif est de colorer les arêtes d’un graphe de telle manière que deux arêtes adjacentes ne reçoivent pas la même couleur. Ce type de coloration prend son origine dans des problèmes de télécommunications : les sommets représentent des tours radio, les arêtes des canaux de communication entre les tours et les couleurs représentent les fréquences utilisées pour communiquer. Dans ce contexte, les contraintes sur les couleurs des arêtes adjacentes peuvent être interprétées comme une condition nécessaire pour éviter les interférences entre deux canaux de communication autour d’une tour radio. Le célèbre Théorème de Vizing [184] nous assure que $\Delta(G) + 1$ couleurs suffisent pour colorer n’importe quel graphe G , où $\Delta(G)$ représente le degré maximum du graphe G (*i.e.* le maximum, pris sur tous les sommets u de G , du nombre d’arêtes incidentes à u).

Nous sommes intéressé par plusieurs généralisations de ce problème. Chaque chapitre de cette seconde partie est consacré à une notion particulière de coloration d’arêtes. Ces généralisations sont de deux types : soit les conditions interdisant que deux arêtes aient la même couleur sont modifiées; soit on souhaite que la coloration d’arêtes nous apporte

des propriétés plus fortes. Le premier type de généralisation est plutôt simple à décrire : on augmente juste le nombre de contraintes entre les couleurs d'arêtes. Par exemple, si on impose que deux arêtes à distance au plus 2 aient des couleurs différentes alors on parle de *coloration forte d'arêtes*. Dans la deuxième famille de colorations que nous étudions, chaque sommet du graphe est associé à une couleur calculée à partir des couleurs de ses arêtes incidentes. Le but est de colorer les arêtes de telle manière que la coloration des sommets soit propre (*i.e.* les sommets adjacents reçoivent des couleurs différentes). Dans ce cas, on parle de *colorations distinguantes*.

Organisation du manuscrit

Le chapitre 1 contient des pré-requis. La plupart des concepts introduits dans ce chapitre sont nécessaires à la compréhension de cette thèse même s'ils n'en sont pas les objets d'étude. En particulier, plusieurs définitions classiques de théorie des graphes et de complexité γ sont présentées.

La première partie de cette thèse « Partie I: Graphes signés » compte quatre chapitres.

Le Chapitre 2 présente les graphes signés et formalise plusieurs notions qui leur sont associées. La plupart des définitions sur les graphes signés sont données dans ce chapitre.

Dans le Chapitre 3, nous étudions la complexité et la complexité paramétrée de certains problèmes concernant les homomorphismes de graphes signés. En particulier, nous répondons à des questions du type : « Combien de sommets/arêtes doit-on enlever à (G, σ) pour qu'il admette un homomorphisme vers (H, π) ». Nous nous intéressons tout particulièrement aux cas où le graphe signé (H, π) est petit et fixé. Dans ces cas-là, admettre un homomorphisme vers (H, π) peut souvent être traduit en une propriété sur le graphe signé donné en entrée. De plus, nous considérons ces questions pour deux types d'homomorphismes de graphes signés différents et nous prouvons des dichotomies de complexité dans chacun des cas.

Le Chapitre 4 est consacré à l'étude du lien entre le nombre chromatique d'un graphe signé et son nombre de cycles à travers un paramètre appelé le *nombre cyclomatique* du graphe. Le nombre cyclomatique d'un graphe G est égal au nombre d'arêtes qu'il faut retirer à G pour enlever tous les cycles du graphe. Nous donnons des bornes supérieures sur diverses notions de nombres chromatiques d'un graphe signé (en incluant des versions liste), qui sont linéaires en le nombre cyclomatique du graphe.

Dans le Chapitre 5, nous étudions l'impact du produit Cartésien sur les graphes signés. Le produit Cartésien des graphes signés est une opération qui prend en entrée deux (ou plusieurs) graphes signés et en crée un nouveau qui est le produit de chacun des facteurs donnés en entrée. Comme n'importe quel produit, nous pouvons nous intéresser à ses propriétés algébriques, et en particulier nous pouvons nous demander si l'on peut écrire n'importe quel graphe signé sous la forme d'un unique produit de facteurs premiers (*i.e.* des facteurs qui ne sont pas eux-mêmes le produit de graphes signés plus petits). Nous démontrons non seulement un théorème de factorisation unique en facteurs premiers mais nous donnons aussi un algorithme permettant de trouver cette factorisation en temps linéaire. Nous étudions aussi les liens entre le produit Cartésien et le nombre chromatique des graphes signés. Nous calculons le nombre chromatique de certains produits et nous en déduisons des bornes supérieures sur le nombre chromatique d'un produit en fonction des nombres chromatiques de ses facteurs.

La deuxième partie de cette thèse « Partie II: Coloration d'arête avec contraintes » compte cinq chapitres.

Le premier chapitre, Chapitre 6, contient une brève introduction aux différentes notions de coloration d'arêtes qui sont étudiées dans les chapitres suivants.

Dans le Chapitre 7, nous étudions un compromis entre la coloration propre d'arêtes et la coloration forte d'arêtes sur les graphes subcubiques. Étant donné un entier t , nous donnons une borne supérieure sur le nombre de couleurs requis pour avoir une coloration propre d'arêtes d'un graphe subcubique telle que deux arêtes à distance au plus 2 ayant la même couleur soient colorées par l'une des t premières couleurs. Les t premières couleurs fonctionnent selon les règles de la coloration propre d'arêtes alors que les autres fonctionnent selon les règles de la coloration forte d'arêtes.

Le Chapitre 8 présente notre étude de la complexité de la coloration injective d'arêtes pour diverses classes de graphes. Une coloration injective d'arêtes est une coloration d'arêtes où pour chaque chemin uvw du graphe, les arêtes uv et wx ne reçoivent pas la même couleur. Nous montrons que calculer le nombre minimum de couleurs nécessaires pour avoir une coloration injective d'arêtes est un problème NP-complet quand le nombre de couleurs est petit (3 ou 4) sur des classes de graphes peu denses. Nous montrons aussi que $O(\sqrt{\Delta(G)})$ couleurs suffisent pour que le problème soit NP-complet. Nous donnons aussi un algorithme FPT pour tester si t couleurs suffisent ou non pour colorer un graphe, paramétré par la largeur d'arborescence du graphe.

Dans le Chapitre 9, nous prouvons que $\Delta(G) + 1$ couleurs suffisent pour avoir une coloration AVD d'un graphe sous différentes conditions. Une coloration AVD est une coloration propre d'arêtes telle que deux sommets adjacents n'aient pas le même ensemble de couleurs sur leurs arêtes incidentes. Nous prouvons ce résultat quand le degré maximum du graphe est significativement plus grand que son degré moyen maximum, ou quand le graphe est planaire et a un degré maximum supérieur ou égal à 12. Ces deux résultats reposent sur deux arguments clés : un algorithme de recoloration qui permet d'enlever des petits sommets voisins et un argument de dénombrement qui permet d'enlever les sommets avec beaucoup de petits voisins.

Finalement, dans le Chapitre 10, nous prouvons la 1-2-3 Conjecture multiplicative : pour tout graphe connexe (non réduit à une arête), on peut colorer ses arêtes avec les couleurs 1, 2 et 3 de telle manière que la coloration (de sommets) obtenue en associant à un sommet le produit des couleurs de ses arêtes incidentes est propre. Nous proposons aussi une conjecture plus faible dans le cas où nous n'avons que deux couleurs pour les arêtes. Enfin, nous étudions la version liste de ce problème dans le cas général ou pour certaines classes de graphes particulières.

Introduction (in English)

The goal of this thesis is to study some graph coloring problems. A *graph* is a mathematical structure composed of a finite number of vertices and a finite number of edges joining pairs of vertices. Graphs are an abstract representation which can be used in diverse domains, for example: road networks, communication networks, communities analysis, databases, DNA analysis, etc.

An interesting use of graphs is in the domain of social psychology which studies the dynamic of relationships between entities. For example, in [7], they use graphs to describe the relationships between countries involved in World War I. For this they use a graph model where edges can be of two types: positive and negative. Positive edges represent friendship while negative edges represent enmity. Such a graph is called a *signed graph* and is generally noted as an ordered pair (G, σ) . Signed graphs were introduced by Harary in [92].

The main ideas behind this concept are as follows. Suppose that you have three countries (or persons) A , B and C , having relationships with each other. Then some situations can be characterized as stable while others can be characterized as unstable. This of course generalizes with more than three countries (see [7]). Figure 2 represents the four possible situations between our three countries.

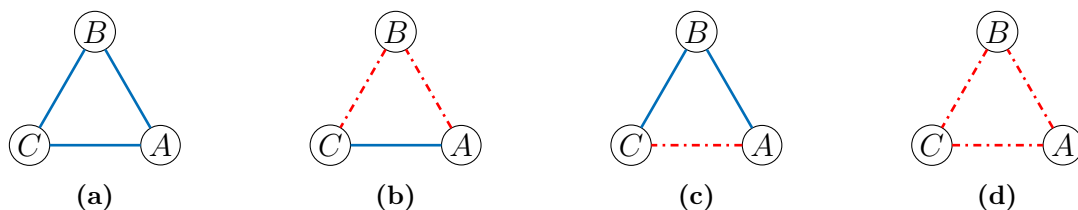


Figure 2: The four possible relationship situations between A , B and C . Positive edges are drawn with full blue lines while negative edges are drawn with dashed red lines.

Let us go into more details.

- The first possible case is when the three countries are all friends (See Figure 2(a)). In this case, the situation is stable. This does not imply that it cannot change but that a change in the relationships does not originate from instabilities in the graph. Such examples of this stable relationship are numerous, for example, one can consider the European Union where every member country is allied with the others.
- The second possible case is when the two countries, say A and C , are allied against the third one (See Figure 2(b)). In this case, the situation is also stable. This is the classical block versus block situation in geopolitics. For example, consider the alliance between the Rohan and the Gondor against the armies of Sauron in *The Lord of the Rings: The Return of the King*.

- The third possible case is when two countries, say A and C , are enemies while the third one, B , is friend with both of them (See Figure 2(c)). This situation is unstable: B is forced to choose between his friendship with A and his friendship with C . Note that it is also possible that A and C reconcile through their common friend. In all cases, this situation is bound to change because of the relationships between A , B and C . For example, in *Star Wars: Episode III – Revenge of the Sith*, Anakin had to choose between Palpatine and the Jedi Order.
- The fourth possible case is when all three countries are enemies (See Figure 2(d)). This situation is also unstable, as the saying goes: “the enemy of my enemy is my friend”. Hence it is likely that two of the three participants will form an alliance against the third. For example, in *Game of Thrones* and in *A Song of Ice and Fire*, the Night’s Watch, the Free Folk and the White Walkers were in a three way war until the two former decided to ally themselves against the latter one in order to survive.

These examples are presented with only three vertices but can be generalized to a greater number of vertices. More generally, it is also possible to consider that A and B do not have a relationship of any kind: in this case, no edge is drawn between A and B on the signed graph.

The notion of “stability” formalized by Harary [92] under the name of *balanced* allows us to study signed graphs. It is interesting to be able to compare signed graphs, to see whether two signed graphs have similar stable and unstable situations. From this follows the notion of homomorphisms. Informally, there is a homomorphism between two signed graphs (G, σ) and (H, π) if we can transfer stable (resp. unstable) cycles of (G, σ) to stable (resp. unstable) cycles of (H, π) . Applying homomorphisms is also a way to do some clustering on the vertices of a signed graph, that is to say to find vertices which behave in a similar fashion.

The study of homomorphisms of signed graphs (and of some derived notions) is the focus of the first part of this thesis.

The second part of this thesis is devoted to edge-coloring of graphs. The goal here is quite simple to describe. We want to color each edge of the graph so that no two adjacent edges receive the same color. This problem takes root in telecommunications: the vertices are radio towers, edges are communication channels between the towers and the colors represent the frequencies used to communicate. In this context, the constraint on the colors can be interpreted as a necessary condition to avoid interference between two communication channels around a tower. A famous theorem by Vizing [184] shows that $\Delta(G) + 1$ colors are sufficient to properly edge-color any graph G , where $\Delta(G)$ is the maximum degree of the graph G (*i.e.* the maximum, taken over every vertex u of G , of the number of edges incident with u).

We are interested into several generalizations of this problem. Each chapter of the second part is devoted to a particular notion of edge-coloring. These generalizations are of two main types: either we change the condition that forbids two edges to be assigned the same color; or we want to obtain stronger properties on the edge-coloring. The first type of generalization is quite simple to describe: we just increase the number of constraints on the edges. For example, if any two edges at distance 2 must receive different colors, then we say our coloring is a *strong edge-coloring* of G . In the second family of generalizations, each vertex is associated with a color computed from the edge-colors of its incident edges.

The goal is to make this vertex coloring proper (*i.e.* no two adjacent vertices receive the same color). We talk about *distinguishing colorings*.

Layout of the manuscript

Chapter 1 contains some prerequisite notions. Most concepts introduced in this chapter are not the focal point of this thesis but are necessary to its understanding. In particular, many classical graph and complexity definitions are introduced in this chapter.

The first part “Part I: Signed graphs” comprises four chapters.

Chapter 2 presents signed graphs in a more complete and formal setting. Most definitions on signed graphs are given in this chapter.

Chapter 3 contains our study of the complexity and parameterized complexity of some problems concerning homomorphisms of signed graphs. In particular we answer questions such as: “how many vertices/edges do we need to remove from (G, σ) so that it admits a homomorphism to (H, π) ”. We mainly focus on cases where the signed graph (H, π) is fixed and is quite small. In these cases, admitting a homomorphism to (H, π) can often be translated to a simple property on the input signed graph. Moreover, we consider these questions for two types of homomorphisms of signed graphs and prove dichotomies for the complexity in each considered case.

In Chapter 4, we study the relationship between the chromatic number of a signed graph and its number of cycles through a parameter called the *cyclomatic number* of a graph. The cyclomatic number of a graph G is simply the number of edges that need to be removed from G so that no cycle remains in G . We give upper bounds on multiple notions of the chromatic number of a signed graph (including some list versions), which are linear in the cyclomatic number of the signed graph.

Chapter 5 contains our study of the impact of the Cartesian product on signed graphs. The Cartesian product of signed graphs is a product operation: it takes two (or more) signed graphs and produce a new signed graph which is the product of each of the factors. Like any product, we can question its algebraic behavior, and in particular whether we can write any signed graph as a product of prime signed graphs (in the sense that these prime signed graphs cannot be written as the product of smaller signed graphs). We not only provide a unique prime factorization theorem for signed graphs but also give an algorithm finding a prime decomposition in linear time. We also study the relationship between the Cartesian product and the chromatic number of signed graphs. We study the chromatic number of some products and derive upper bounds on the chromatic number of a product depending on the chromatic numbers of each of its factors.

The second part “Part II: Edge-coloring with constraints” comprises five chapters.

Chapter 6 gives a brief introduction to the different notions of edge-coloring that are studied in the following chapters.

In Chapter 7, we study a compromise between two types of edge-coloring on subcubic graphs. Given a number t , we give an upper bound on the number of colors required to properly edge-color a subcubic graph so that, except for the first t colors, any pair of edges with the same color are at distance at least 3 in the graph. This coloring notion is a mixture of proper edge-coloring and strong edge-coloring. Indeed, the first t colors behave according to the rules of proper edge-coloring while the remaining colors behave according to the rules of strong edge-coloring.

Chapter 8 presents our study of the complexity of injective edge-coloring on various classes of graphs. An injective edge-coloring is an edge-coloring where for any path uvw of the graph, the edges uv and wx do not have the same color. We show NP-completeness when the number of colors is small (3 or 4) on some sparse classes of graph. We also show that $O(\sqrt{\Delta(G)})$ colors are sufficient for the problem to be NP-complete. We also provide an FPT algorithm for testing whether t colors are sufficient or not for some graph, parameterized by the treewidth of the input graph.

In Chapter 9, we prove that $\Delta(G) + 1$ colors are sufficient in order to have an adjacent vertex distinguishing (AVD) coloring a graph under sufficient conditions. An AVD coloring is a proper edge-coloring for which no two adjacent vertices have the same set of colors on their incident edges. We prove this result when the maximum degree $\Delta(G)$ of a graph is significantly greater than its maximum average degree and when the graph is planar and has maximum degree at least 12. Both results rely on two clever arguments: a recoloring algorithm which allows us to remove small adjacent vertices and a counting argument which allows to remove vertices with many small neighbors.

Finally in Chapter 10, we prove the Multiplicative 1-2-3 Conjecture, *i.e.* that every connected graph (which is not just an edge) can be edge-colored with colors 1, 2 and 3 so that the coloring of G , obtained by associating with each vertex the product of the colors on edges incident with u , is proper. We also propose a weaker conjecture when only the labels 1 and 2 are available. Finally, we study the list version of the problem for general graphs and for particular classes of graphs.

Chapter 1

Preliminaries

This chapter presents general concepts that are necessary to the understanding of this thesis.

In Section 1.1, we present the core concept of this thesis: graphs. We introduce notation and present the basic definitions of Graph Theory. We also present some famous graph parameters and important graph classes.

In Section 1.2, we present the notion of graph coloring. We first focus on vertex coloring and its links with the notion of homomorphism. We then present other various ways of coloring a graph.

In Section 1.3, we introduce the fundamental notions of Complexity Theory required to understand this thesis. We most notably present the notion of NP-completeness and the notion of parameterized complexity.

Finally, in Section 1.4, we recall some classical algebra concepts. Among the four, this section is the least central in this thesis but it might shed some light on some particular sections of the following chapters.

Contents

1.1	Graphs: definitions, notation	18
1.1.1	Other types of graphs	18
1.1.2	Subgraphs and some important subgraphs	20
1.1.3	Walks	21
1.1.4	Applications of paths: connectivity and distance in graphs	23
1.1.5	Bipartite graphs	24
1.1.6	Line graphs	24
1.1.7	Graph classes and parameters related to the degrees of vertices	25
1.1.8	Planar graphs	27
1.1.9	Treewidth	29
1.2	Colorings and Homomorphisms	30
1.2.1	Vertex coloring	31
1.2.2	Homomorphisms	33
1.2.3	Homomorphisms on more complex graphs	34
1.2.4	Edge-coloring	35
1.2.5	List coloring	37

1.3	NP-completeness and FPT	38
1.3.1	Basics of Complexity Theory	38
1.3.2	Algorithmic representations of graphs	39
1.3.3	NP-completeness	40
1.3.4	Parameterized complexity	43
1.3.5	Some problems used in reductions	45
1.4	A little bit of algebra	46
1.4.1	Well-founded ordering and two classical proof methods in Graph Theory	46
1.4.2	Equivalence and quotient	47
1.4.3	Algebraic structures	48
1.4.4	Combinatorial Nullstellensatz	49

1.1 Graphs: definitions, notation

We generally follow terminology and notation of [191].

An *undirected graph* G is a triplet $(V(G), E(G), \varphi)$ where $V(G)$ and $E(G)$ are two disjoint sets, the set of vertices and the set of edges respectively. Here φ is an *incidence relation* which associate to each edge two incident vertices called its *endpoints*. In this thesis, the term “graph” refers to an undirected graph.

A *loop* in a graph G is an edge whose endpoints are equal. We often consider graphs without loops, called *loopless graphs*. If there are edges with the same endpoints u and v in a graph G , we say that uv is a *multi-edge* or that the edges between u and v are *multiple edges*. The *multiplicity* of a multi-edge uv is the number of edges incident with both u and v . An undirected graph is *simple* if it does not have loops nor multiple edges. To avoid confusion, we use the term *multi-graph* for a graph which can have multiple edges. Note that most graphs in this thesis are simple.

For simplicity, we often “forget” the incidence relation of a graph when we have no multiple edges. An edge e incident with u and v is treated as a pair $\{u, v\}$, denoted uv for concision, where u and v are two vertices of G . An undirected graph is then noted $(V(G), E(G))$ where $E(G) \subseteq V(G)^2$. Note that we often abuse the notation $(V(G), E(G))$ to also refer to multi-graphs, in this case, the incidence relation is implicit.

The *order* of G is $|V(G)|$ and its *size* is $|E(G)|$. A graph with no edges is an *empty graph*. The graph with no vertices nor edges is the *null graph* G_\emptyset .

Two vertices u and v of a graph G are *adjacent* when uv is an edge of G . Two edges e and e' of a graph G are *adjacent* when e and e' have a common endpoint. The *neighborhood* $N_G(u)$ of a vertex u in the graph G is the set of vertices adjacent to u in G . When the graph G is clear from the context, we note $N(u)$ for the neighborhood of u in G . A vertex with no neighbors is an *isolated vertex*.

In a loopless graph, the *degree* of a vertex u in G , denoted $d_G(u)$ or simply $d(u)$, is the number of edges incident with u . If the graph is simple then the degree of u is also the number of neighbors of u , $|N(u)|$. When loops are allowed, the degree of a vertex u is the number of times u is an endpoint of an edge of G . In other words, a loop count twice

for the degree of a vertex. A k -vertex (resp. k^- -vertex, resp. k^+ -vertex) of G is a vertex of degree k (resp. at most k , resp. at least k). A k -neighbor (resp. k^- -neighbor, resp. k^+ -neighbor) of a vertex u is a k -vertex (resp. k^- -vertex, resp. k^+ -vertex) belonging to $N(u)$.

Let us see an example. Let P be the Petersen graph drawn in Figure 1.1. The graph P has order 10 and size 15. The neighborhood of the vertex a_0 is the set $\{b_0, a_2, a_3\}$. In this graph all vertices have degree 3.

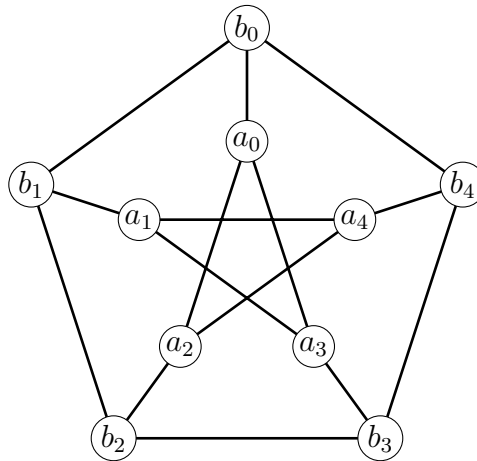


Figure 1.1: A simple graph: the Petersen graph.

An *isomorphism* from G to H is a bijection φ from $V(G)$ to $V(H)$ such that for every vertices x and y of G , $xy \in E(G)$ if and only if $\varphi(x)\varphi(y) \in E(H)$. In this case, we note $G = H$. An *automorphism* of G is an isomorphism from G to G . In general, a graph can have multiple automorphisms. In the rest of this thesis, we consider graphs up to isomorphism (see also Section 1.4.2).

1.1.1 Other types of graphs

There are multiple ways to define a graph. These different varieties of graphs are all dependant on the definition of $E(G)$.

As mentioned before, depending on our definition of the set of edges, we can allow loops and/or multiple edges in our graphs. See Figure 1.2(a) for an example of graph with loops and multiple edges.

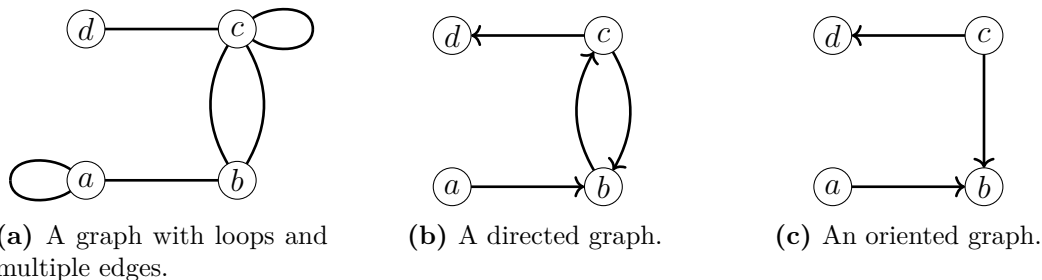


Figure 1.2: Different varieties of graphs.

Another type of graphs is the notion of *directed graphs*, where $E(G)$ is replaced by $A(G)$, the set of *arcs*. The set $A(G)$ is the set of ordered pairs (instead of pairs) (u, v)

where u and v are two vertices. An arc (u, v) can also be noted \vec{uv} . An *oriented graph* is a simple directed graph G with the extra condition that if $\vec{uv} \in A(G)$ then $\vec{vu} \notin A(G)$. In other terms an oriented graph is a directed graph where each pair of vertices can have at most one arc between them. Oriented graphs can be obtained from a simple undirected graph by choosing an orientation for each edge. See Figure 1.2(b) and Figure 1.2(c) for examples of such graphs. Of course it is possible to combine the notion of directed graphs with the notion of loops and multi-edges. For a directed graph G , the *indegree* of a vertex u is the number of arcs of the form \vec{vu} in $A(G)$, and the *outdegree* of a vertex u is the number of arcs of the form \vec{uv} in $A(G)$.

A *mixed graph* G is a triplet $(V(G), E(G), A(G))$ where $V(G)$ is the set of vertices of G , $E(G)$ is the set of edges of G and $A(G)$ is the set of arcs of G . Informally, a mixed graph is a graph with edges and arcs. One can consider such graphs to be simple, in which case they have no loops and for every two vertices u and v of G , G does not contain both the edge uv and the arc \vec{uv} .



Figure 1.3: Graphs with multiple edge/arc types.

A *k-edge-colored* graph is a graph G where the set of edges is partitioned into k sets $E_1(G), \dots, E_k(G)$. We note such a graph $(V(G), E_1(G), \dots, E_k(G))$. The set $E_i(G)$ is the set of edges colored i in G . See Figure 1.3(a) for an example of a 3-edge-colored graph. Note that we will mainly work with 2-edge-colored graphs in this thesis. By default, the two colors used in a 2-edge-colored graph will be the color blue and the color red.

A (m, n) -*mixed* graph is a graph G with m arc types and n edge types. We note such a graph $(V(G), A_1(G), \dots, A_m(G), E_1(G), \dots, E_n(G))$. The set $E_i(G)$ is the set of edges colored i in G and the set $A_j(G)$ is the set of arcs colored j in G . See Figure 1.3(b) for an example of a (2,2)-mixed graph. Note that a (0,1)-mixed graph is an undirected graph, a (1,0)-mixed graph is a directed graph, a (1,1)-mixed graph is a mixed graph and a (0, k)-mixed graph is a k -edge-colored graph.

1.1.2 Subgraphs and some important subgraphs

In this section, we present the notion of subgraph.

Definition 1.1 (Subgraphs). Let H and G be two graphs. We say that H is a *subgraph* of G if and only if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Let X be a subset of vertices of G , the *induced subgraph* $G[X]$ of G is the subgraph of G on vertex set X and where, for every two vertices u and v of X , $uv \in E(G[X])$ if and only if $uv \in E(G)$. If H is a subgraph of G and $H \neq G$, then H is a *proper* subgraph of G .

The notion of subgraph is important in Graph Theory as it formalizes the notion of “containing a particular motif”. It also allows us to focus on particular parts of a graph. See Figure 1.4 for some examples of subgraphs.

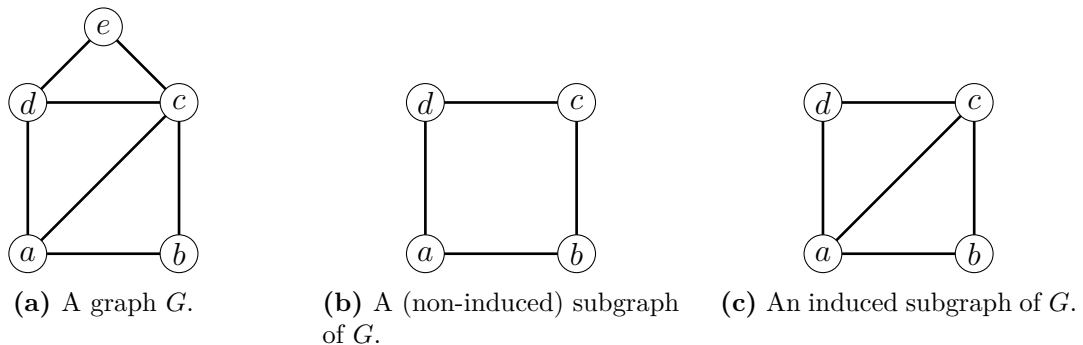


Figure 1.4: A graph (a), and one of its non-induced subgraphs (b) and one of its induced subgraphs (c).

Note that we can create subgraphs by removing edges and vertices from a graph G .

Definition 1.2 (Deleting vertices and edges). Let G be a graph and S_v (resp. S_e) be a subset of vertices (resp. edges) of G . The graph $G - S_v$ is the subgraph of G obtained from G by removing the vertices of S_v and the edges incident with any vertex of S_v . The graph $G - S_e$ is the subgraph of G obtained from G by removing the edges of S_e .

Let us see some important subgraphs that a graph G can contain.

Definition 1.3 (Complete graph, clique, independent set). The *complete graph* K_p is the graph of order p such that for every pair of distinct vertices u and v of K_p , uv is an edge of K_p . If H is a subgraph of G isomorphic to K_p , then we say that H is *clique* (of order p) of G . An *independent set* of order p in a graph G is a set of p vertices which are pairwise non-adjacent.

In some sense, a clique is the “opposite” of an independent set, the former is a subgraph with all possible edges while the latter is a subgraph with no edges. This idea can be formalized with the notion of complement graph.

Definition 1.4 (Complement graph). Let G be a graph. The *complement* of G is the graph, denoted \overline{G} , defined by $V(\overline{G}) = V(G)$ and for any two distinct vertices u and v of G , $uv \in E(G)$ if and only if $uv \notin E(\overline{G})$. Note that the complement of \overline{G} is the graph G itself.

Formally, G has an independent set of order p if and only if \overline{G} has a clique of order p . These two notions allow us to define two graph parameters.

Definition 1.5 (clique number, independent number). The *clique number* of G , denoted $\omega(G)$, is the order of the largest clique of G . The *independence number* of G , denoted $\alpha(G)$, is the order of the largest independent set of G .

Note that $\alpha(G) = \omega(\overline{G})$ for every graph G . See Figure 1.5 for some examples.

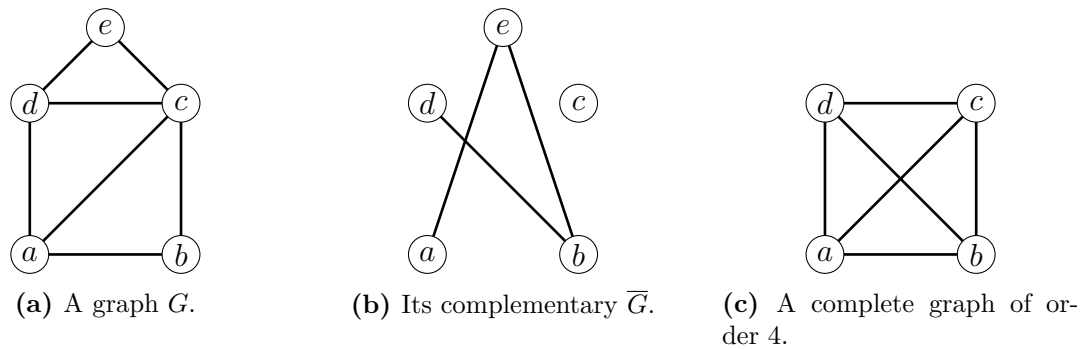


Figure 1.5: A graph G with $\alpha(G) = 2$ and $\omega(G) = 3$ (a), its complement \bar{G} (b) and a complete graph of order 4 (c).

1.1.3 Walks

Another important substructure in a graph is the notion of walks which allows us to traverse a graph.

Definition 1.6 (Walk, closed walk). A *walk* in a graph G is a list s_0, \dots, s_n of vertices of G such that $s_i s_{i+1} \in E(G)$ for every i in $\llbracket 0, n \rrbracket$. An (s_0, s_n) -*walk* is a walk whose first vertex is s_0 and whose last vertex is s_n . More generally, if A and B are two sets of vertices, then an (A, B) -*walk* is an (a, b) -walk for some vertices $a \in A$ and $b \in B$.

A *closed walk* is a walk with $s_0 = s_n$. An *internal vertex* of a walk is a vertex of the walk which is neither the first nor the last vertex of the walk.

The *length* (number of edges, counted with multiplicity) of a walk $W = s_0, \dots, s_n$ is n , and its *order* (number of vertices, counted with multiplicity) is n if W is a closed walk, or $n + 1$ otherwise. When no confusion is possible, we may write $s_0 s_1 \dots s_n$ for the sequence s_0, s_1, \dots, s_n of a walk.

Note that a walk in G is not *stricto sensu* a subgraph, it is a sequence of vertices. Among walks, paths and cycles play an important role.

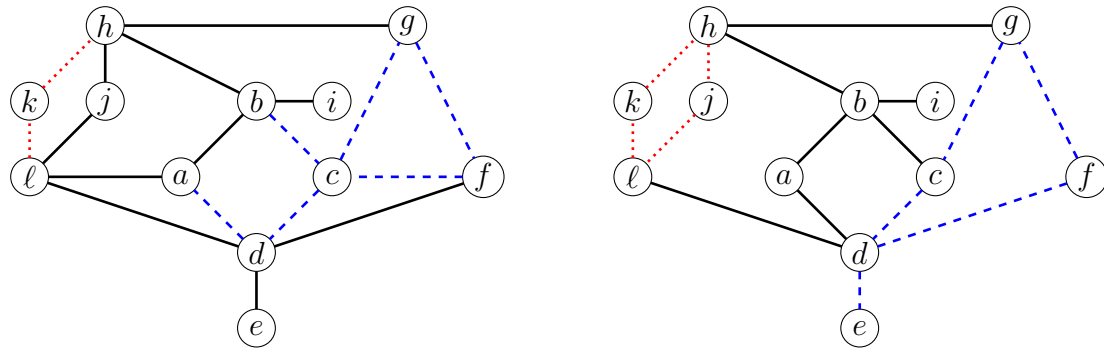
Definition 1.7 (Path, cycle). If all vertices of a walk are pairwise distinct, then the walk is a *path*. An (s_0, s_n) -*path* is a (s_0, s_n) -walk which is a path. If A and B are two sets of vertices, then an (A, B) -*path* is an (a, b) -path for some vertices $a \in A$ and $b \in B$. A closed walk where all vertices are pairwise distinct, except s_0 and s_n , is a *cycle*.

We may consider the sequence of vertices s_0, \dots, s_k of G which is a path (resp. cycle) as the subgraph formed by these vertices and the edges of the form $s_i s_{i+1}$ for $i \in \llbracket 0, k - 1 \rrbracket$. The path of order k is noted P_k . The cycle of order k is noted C_k . An *even cycle* (resp. *odd cycle*) is a cycle of even length (resp. odd length). A cycle of length 3 is also called a *triangle*.

With the notion of cycle, we can define another important graph parameter.

Definition 1.8 (Girth). The *girth* of a graph G is the length of a smallest induced cycle of G .

See Figure 1.6 for examples of a walk, a closed walk, a path and a cycle in graphs of girth 3 and 4.



(a) A graph G of girth 3 with a path hkl (represented in red dotted edges) and a walk $adcfgeb$ (represented in blue dashed edges).
 (b) A graph G of girth 4 with a cycle $hjdk$ (represented in red dotted edges) and a closed walk $edfgcde$ (represented in blue dashed edges).

Figure 1.6: Examples of a walk, a closed walk, a path and a cycle in graphs of girth 3 and 4.

1.1.4 Applications of paths: connectivity and distance in graphs

With the notion of paths (or walks), we can define the notion of connected graphs.

Definition 1.9 (Connected graph, connected component). A graph is *connected* if there exists a path between u and v for every pair of vertices u and v of G . A *connected component* of G is a maximal connected subgraph of G . A graph is *disconnected* if it has at least two connected components. An *cut-vertex* is a vertex whose removal from the graph increases the number of connected components.

A stronger notion is the notion of k -connected graphs.

Definition 1.10 (k -connected graph, connectivity). A graph is *k -connected* if for every pair of vertices u and v of G , there exist k paths, with disjoint internal vertices, between u and v . The *connectivity* of a graph G is the largest integer k for which G is k -connected. A 2-connected graph is also called a *biconnected graph*. A *k -connected component* of G is a maximal k -connected subgraph of G .

See Figure 1.7 for some examples of connectivities.

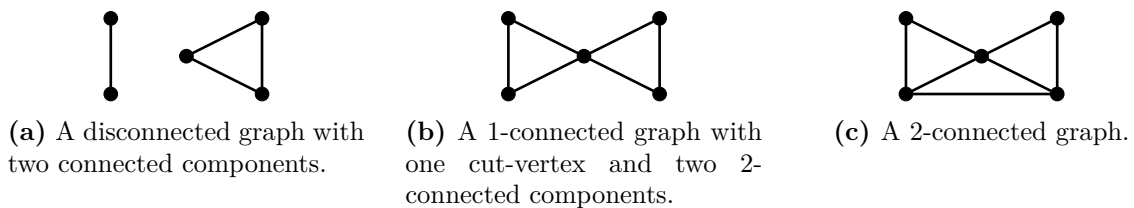


Figure 1.7: Graphs with different connectivities.

Definition 1.11 (Edge cut). For a graph G and $X, Y \subseteq V(G)$, let us note $E(X, Y)$ for the set of edges with one endpoint in X and the other endpoint in Y . An *edge cut* is an edge set of the form $E(X, V(G) \setminus X)$.

Note that for every graph G , if X is a proper non-empty subset of $V(G)$ (i.e. $\emptyset \neq X \subsetneq V(G)$) then the graph $G - E(X, V(G) \setminus X)$ is disconnected.

Another useful application of paths is the notion of distance.

Definition 1.12 (Distance). Let u and v be two vertices of G . The *distance* between u and v , denoted $d_G(u, v)$ is the length of a shortest path connecting u and v . If there is no path between u and v , then $d_G(u, v) = +\infty$. When the context is clear, we note $d(u, v)$ instead of $d_G(u, v)$.

From this notion, we can define the concept of graph powers.

Definition 1.13 (Graph power). Let G be a graph and k be an integer. The k -th power of G , denoted G^k , is the graph with vertex set $V(G)$ in which two vertices are adjacent if and only if they are at distance at most k . The graph G^2 is also called the *square* of G .

In particular, G^0 is an independent set and $G^1 = G$. See Figure 1.8 for one example of distance between vertices and one example of square graph.

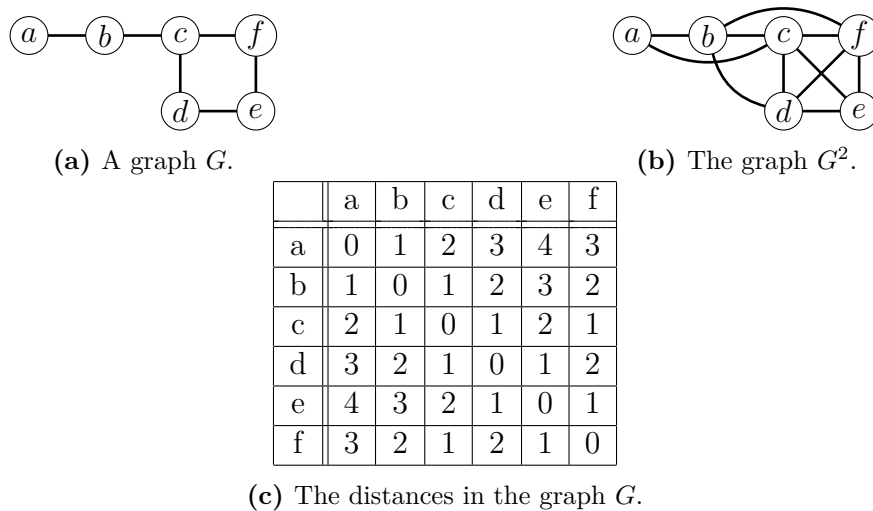


Figure 1.8: Example of distances in a graph and its square graph.

1.1.5 Bipartite graphs

The notion of cycle help us characterize one of the most studied class of graph: bipartite graphs.

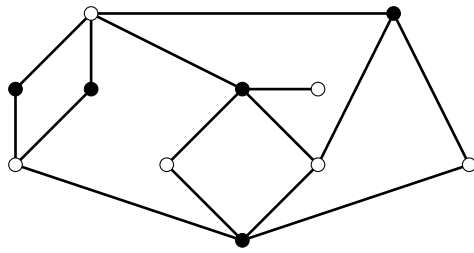
Definition 1.14 (Bipartite graph, bipartite complete graph). A graph G is *bipartite* if we can partition $V(G)$ into $A \uplus B$ (where \uplus is the disjoint union operation) such that each edge xy of G has one endpoint in A and one endpoint in B . In particular, bipartite graphs are exactly graphs without odd cycles.

The *complete bipartite graph* $K_{p,q}$ is the graph composed of two sets of vertices: A of size p and B of size q . The sets A and B are independent sets and for every $u \in A$ and every $v \in B$, $uv \in E(K_{p,q})$.

See Figure 1.9 for two examples of bipartite graphs.

1.1.6 Line graphs

Let us continue with one more type of graph constructed from a graph G , the line graph, which is the graph of relationship between the edges of G .



(a) A bipartite graph.

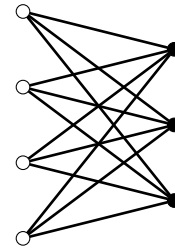
(b) The bipartite graph $K_{4,3}$.

Figure 1.9: Examples of bipartite graphs. For both graphs, the set of vertices is partitioned into the set of white vertices and the set of black vertices.

Definition 1.15 (Line graph). The *line graph* $L(G)$ of a graph G is the graph whose vertex set is the set of edges of G and for which two vertices of $L(G)$ (*i.e.* two edges of G) are adjacent in $L(G)$ if and only if they are adjacent as edges in G . The *degree of an edge* e of G is the degree of the vertex corresponding to e in $L(G)$. Alternatively, it is the number of edges adjacent to e in G .

See Figure 1.10 for an example of a line graph.

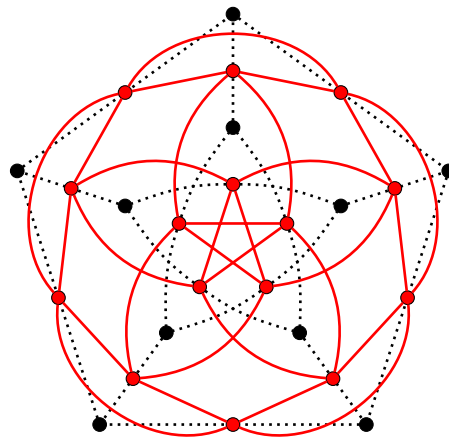


Figure 1.10: The line graph (represented in red) of the Petersen graph (represented in black).

We can also define a notion of distance between edges of a graph G using $L(G)$.

Definition 1.16 (Distance of edges). Let e and e' be two edges of G . The *distance* between e and e' , denoted $d_G(e, e')$ is the length of a shortest path connecting e and e' in the line graph $L(G)$. If there is no path between e and e' , then $d_G(e, e') = +\infty$. The distance between e and e' can also be defined as the smallest number of vertices of a path v_1, \dots, v_k of G such that $v_1 \in e$ and $v_k \in e'$.

Note that the distance between two edges e and e' of a graph G could also have been defined as the minimum of $d_G(a, b)$ for $a \in e$ and $b \in e'$.

1.1.7 Graph classes and parameters related to the degrees of vertices

Let us start by presenting some important graph parameters.

Definition 1.17 (Minimum degree, maximum degree). Let G be a graph. The *minimum degree* of G , denoted $\delta(G)$, is the minimum of the degrees $d(u)$ where u is a vertex of G . The *maximum degree* of G , denoted $\Delta(G)$, is the maximum of the degrees $d(u)$ where u is a vertex of G .

For example, the minimum degree of a path is 1 while the maximum degree of a path of length at least 2 is 2. The Petersen graph (see Figure 1.1) has minimum degree and maximum degree 3.

The notion of degree can help us characterize some graph classes.

Definition 1.18 (Regular graphs). A graph is *regular* if all of its vertices have the same degree: $\delta(G) = \Delta(G)$. A 3-regular graph is a *cubic graph*. A graph G with $\Delta(G) \leq 3$ is a *subcubic graph*.

Another important class is the following one.

Definition 1.19 (Degeneracy). A graph G is *d-degenerate* if all of its subgraphs contain a vertex of degree at most d .

A particular class of degenerated graphs is the class of forests.

Definition 1.20 (Forests and trees). A *forest* F is a 1-degenerate graph. Alternatively, a forest is an *acyclic* graph (*i.e.* a graph with no cycle). A *tree* is a connected forest. A *leaf* of a forest is a vertex of degree 1. A subgraph of a graph G which is isomorphic to a forest (resp. to a tree) is a *subforest* of G (resp. a *subtree* of G).

A important type of subforest is the following one.

Definition 1.21 (Spanning forest, spanning tree). Let G be a graph (resp. a connected graph). A *spanning forest* is a subforest of G on vertex set $V(G)$ which has the same number of connected components as G . A *spanning tree* is a spanning forest of a connected graph.

See Figure 1.11 for an example of spanning tree.

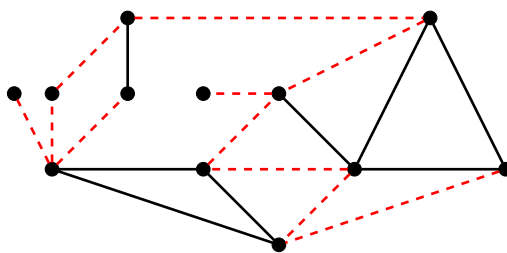


Figure 1.11: A graph G and one of its spanning trees represented with red dashed edges.

Another notion related to the degree is the notion of average degree.

Definition 1.22 (Average degree, maximum average degree). The *average degree* of a graph G , denoted $\text{ad}(G)$, is the average of the degrees of the vertices of G :

$$\text{ad}(G) = \frac{2|E(G)|}{|V(G)|}$$

The *maximum average degree* of a graph G , denoted $\text{mad}(G)$, is the maximum of $\text{ad}(H)$ taken over all subgraphs H of G .

The notion of average degree allows us to determine whether a graph is dense or not. If $\text{ad}(G) = o(|V(G)|)$, then the graph G is *sparse*, otherwise it is *dense*.

Definition 1.23 (Hereditary property, hereditary class). A graph property is *hereditary* if for every G which verifies the property, the property is true for all subgraphs of G . A graph class is *hereditary* if for every G in the class, all subgraphs of G also belong to the class.

Note that the average degree is not hereditary, for example even if the average degree of the graph composed of a clique of order p and $(p - 1)^2$ isolated vertices has average degree 1, it contains a subgraph of average degree $p - 1$. This is why the notion of maximum average degree is important: an upper bound on $\text{mad}(G)$ is also valid for the maximum average degree of any subgraph of G . See Figure 1.12 for an example of these parameters.

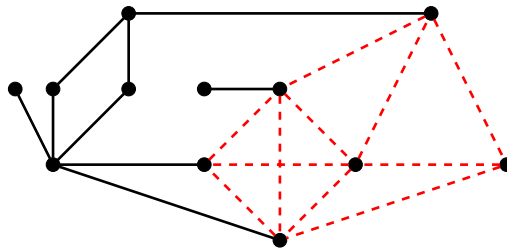


Figure 1.12: A graph G of minimum degree 1, maximum degree 5, average degree $\frac{10}{3}$ and maximum average degree $\frac{11}{3}$. The subgraph of G achieving an average degree of $\frac{11}{3}$ is represented with red dashed edges.

Note that a d -degenerate graph G has at most dn edges and thus has average degree at most $2d$. Moreover, the class of d -degenerate graphs is hereditary, hence if G is d -degenerate, then $\text{mad}(G) \leq 2d$. Conversely, if $\text{mad}(G) < k$ then G is $(k - 1)$ -degenerate.

1.1.8 Planar graphs

One particularly interesting class of graphs is the class of planar graphs. To define them, we first need the notion of planar embedding.

Definition 1.24 (Drawing of a graph, planar embedding). A *drawing* M of a graph G is a function which associate with each vertex v a coordinate $M(v)$ in \mathbb{R}^2 and which associate with each edge uv an injective continuous function $M(uv) : [0, 1] \rightarrow \mathbb{R}^2$ such that $\{M(0), M(1)\} = \{M(u), M(v)\}$.

Moreover, the coordinates of the vertices are distinct, and if uv is an edge and w is a vertex different from u and v , then $M(w)$ does not belong to the embedding of the edge uv . Finally, for every two edges e_1 and e_2 and every two real numbers t_1 and t_2 with $0 < t_1 < 1$ and $0 < t_2 < 1$, if $M(e_1)(t_1) = M(e_2)(t_2)$ then this point is a *crossing*.

A *planar embedding* is a drawing of a graph without crossings.

Definition 1.25 (Planar graph). A *planar graph* is a graph G which admits a planar embedding M . A *plane graph* is a planar graph G with a particular planar embedding M of G .

See Figure 1.13 for an example of a planar graph. From Figure 1.13, one can see that a planar graph may be drawn on the plane with crossings. Moreover, note that a planar graph can have multiple non-topologically equivalent planar embeddings. For example, in Figure 1.13(c), placing the vertex e in between a and b yield another planar embedding which is not topologically equivalent to the one in the figure.

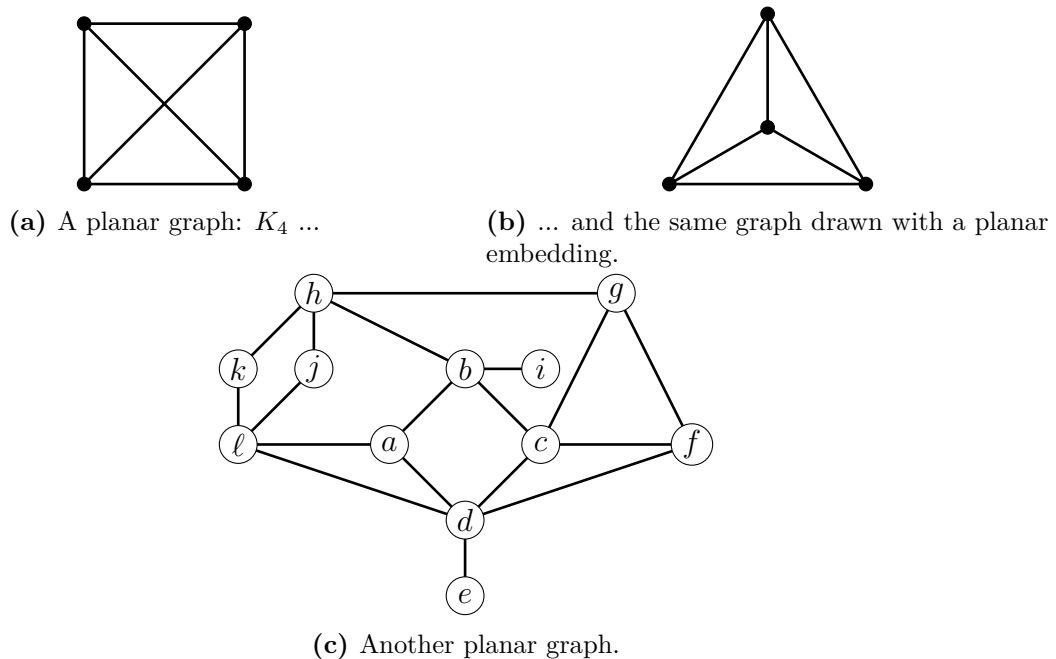


Figure 1.13: Two planar graphs.

Of course, not all graphs are planar. Two minimal (for the subgraph relation) non-planar graphs are K_5 and $K_{3,3}$ (see Figure 1.14).



Figure 1.14: Two non-planar graphs.

In fact, these two graphs are essential to the study of planar graphs. A *subdivision* of a graph G is a graph obtained from G by replacing the edges of G by arbitrary paths of length at least 1.

Theorem 1.26 (Kuratowski's Theorem [135]). *A graph is planar if and only if it does not contain a subgraph which is a subdivision of K_5 or $K_{3,3}$.*

Also note that it is possible to determine in time $O(|E(G)|)$ whether G is planar or not thanks to Kuratowski's Theorem [194].

Definition 1.27 (Face). Let G be a planar graph together with a planar embedding M . A *face* of G in M is a maximal connected region of the plane which does not contain points of the embedding (*i.e.* no image of the vertices nor the curves which represent the edges). The *outer face* is the only unbounded face of G in M . Let us note $F(G, M)$ the set of faces of G in M .

A face f is *incident* with a vertex v (resp. an edge e) if and only if $M(v)$ (resp. the image of $M(e)$) belongs to (resp. is included in) the boundary of f , *i.e.* $M(v)$ (resp. the image of $M(e)$) belongs to (resp. is included in) the closure of f . Two faces are *adjacent* if and only if they are incident with a common edge e . The *length* or *degree* of a face f , denoted $d(f)$, is the number of edges incident with f counted with multiplicity (*i.e.* if an edge is incident with only one face, we count this edge twice for this face). We often note a face by the ordered list of its vertices.

The graph in Figure 1.13(c) has eight faces: $abcd$, adl , cdf , cgf , $bcghbi$, $abhjl$, $ljhk$ and the unbounded face $lkhgfded$.

A well known property of planar graphs is given by Euler's formula, which connects the number of vertices, the number of edges and the number of faces of a graph.

Theorem 1.28 (Euler's Formula). *Let G be a planar graph with a planar embedding M . We have:*

$$|V(G)| - |E(G)| + |F(G, M)| = 2.$$

In particular the number of faces of G in M does not depend on the planar embedding M .

This result can easily be shown by induction on the number of vertices. By using the fact that $\sum_{v \in V(G)} d(v) = 2|E(G)| = \sum_{f \in F(G, M)} d(f)$, one can note that Euler's Formula is equivalent to the following identity:

$$\sum_{v \in V(G)} (d(v) - 6) + \sum_{f \in F(G, M)} 2(d(f) - 3) = -12. \quad (1.1)$$

A classical way to obtain contradictions on planar graphs is to suppose the existence of a planar graph verifying some property and then showing that this planar graph does not follow Euler's Formula. This implies that no planar graph verifies the supposed property.

Note that Euler's Formula give a relationship between the girth and the maximum average degree of a planar graph when observing that $2|E(G)| \geq g|F(G, M)|$.

Proposition 1.29. *If G is a planar graph with girth g then $\text{mad}(G) < \frac{2g}{g-2}$.*

Also note that Equation (1.1) implies that a simple planar graph is 5-degenerate.

1.1.9 Treewidth

Another important graph parameter of Graph Theory that we use in the context of parameterized complexity (see Section 1.3), is the notion of treewidth.

Definition 1.30 (Rooted tree). A *rooted tree* is a tree T where we distinguish one vertex called the *root* of T . In a rooted tree, if uv is an edge and $d(r, u) < d(r, v)$, then we say that u is the *parent* of v in T , or that v is the *child* of u in T . A vertex of a rooted tree is often called a *node*. The *subtree of T rooted at u* is the rooted tree obtained from T by removing the edge between u and its parent and keeping only the vertices which are in the same connected components as u .

The notion of treewidth is defined as follows.

Definition 1.31 (Tree decomposition, treewidth). A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, (X_u)_{u \in V(T)})$ where T is a rooted tree and where, for every node u of T , X_u is a subset of $V(G)$ associated with u , called the *bag* of u , verifying the following properties:

1. for every vertex v of G , there is a node u of T such that $v \in X_u$,
2. for every vertex v of G , the set of nodes whose bag contains v induces a subtree of T ,
3. for every edge vw of G , there is a node u whose bag contains both v and w .

The *width* $w(T)$ of the tree decomposition T is the largest size of a bag X_u , for $u \in V(T)$, minus 1:

$$w(T) = \max_{u \in V(T)} |X_u| - 1.$$

The *treewidth* $tw(G)$ of a graph G is the smallest width of a tree decomposition of G .

See Figure 1.15 for an example of a tree decomposition.

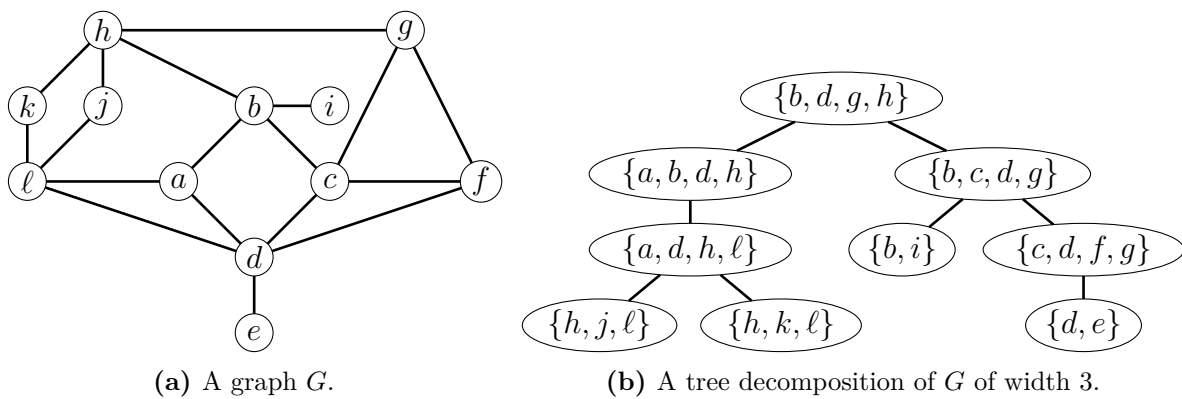


Figure 1.15: A graph and a tree decomposition of this graph.

Let us see some examples: a forest F has treewidth 1, the complete graph K_p has treewidth $p - 1$ and a cycle has treewidth 2.

Definition 1.32 (Grid). A *grid* $G_{n,m}$ is a graph on nm vertices $(v_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$ where $v_{i,j}$ and $v_{k,\ell}$ are adjacent if and only if $i = k$ and $|j - \ell| = 1$, or $j = \ell$ and $|i - k| = 1$.

This famous theorem from Robertson and Seymour shows that planar graphs have unbounded treewidth.

Theorem 1.33 ([167]). *For every integer k , there is a grid G with $tw(G) > k$.*

One can add constraints to the tree decomposition to make it easier to work with. Nice tree decompositions [130] are a well-known tool for designing algorithms on graphs of bounded treewidth using dynamic programming.

Definition 1.34 (Nice tree decomposition). A *nice tree decomposition* of a graph G is a tree decomposition T , rooted at a node $Root$. Each node of T is of one of the following types.

- A *join node* has exactly two children, with the same bags as their parent join node.
- An *introduce node* has a unique child and contains exactly one more vertex in its bag than its child's bag.
- A *forget node* also has a unique child, but the forget node's bag has exactly one less vertex than its child's bag.
- A *leaf node* is a leaf of the tree and contains no vertices.

This type of decomposition can be obtained from a tree decomposition in polynomial time and will be useful in Chapter 8.

1.2 Colorings and Homomorphisms

Each chapter of this thesis will revolve around the idea of coloring graphs. Depending on the type of graphs, we may define multiple notions of coloring. In this section, we first present the classical notion of vertex coloring, and then the notion of homomorphism of graphs, which gives another point of view on vertex coloring. Finally we describe other classical notions of coloring: oriented coloring, vertex coloring of (n, m) -mixed graphs, edge-coloring and list coloring.

Note that each part/chapter of this thesis focuses on a particular coloring notion which will be described in the corresponding section.

1.2.1 Vertex coloring

Our first notion of coloring is the notion of vertex coloring.

Definition 1.35 (Vertex coloring and chromatic number). A *vertex k -coloring* of a graph G is a function from $V(G)$ to the set of colors $\llbracket k \rrbracket$. A vertex k -coloring of a graph G is *proper* if no two adjacent vertices receive the same color.

The *chromatic number* $\chi(G)$ of a graph G is the smallest k such that G admits a proper vertex k -coloring.

Note that a proper vertex coloring is ill-defined on graphs with loops as a vertex is adjacent to itself. Also note that multiple edges do not matter here: if u and v are adjacent, then it does not matter whether there are 1, 2 or 10 edges between them for the coloring. This is why we often only consider proper vertex coloring on simple graphs. Also note that a simple graph G always admits a proper vertex $|V(G)|$ -coloring, hence the chromatic number of a simple graph is always finite.

Remark. Unless stated otherwise, in this thesis, a k -coloring always refer to a vertex k -coloring.

Even if formally our colors are elements of $\llbracket k \rrbracket$, when drawing a coloring we associate with each element of $\llbracket k \rrbracket$ an RGB color. See Figure 1.16 for an example of a proper vertex 3-coloring. The proper vertex 3-coloring in Figure 1.16 implies that the chromatic number of the Petersen graph is at most 3.

Let us collect some easy facts on the chromatic number.

Proposition 1.36 (Folklore). *Let G be a graph. The following statements are true.*

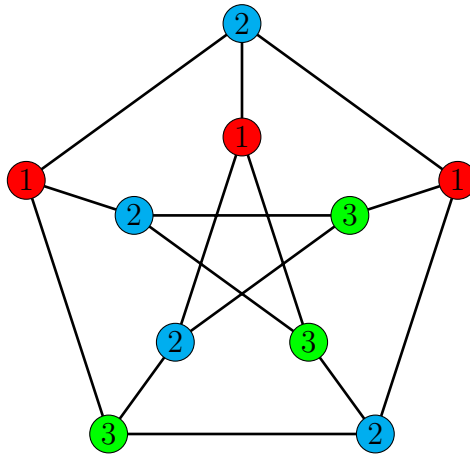


Figure 1.16: A proper vertex 3-coloring of the Petersen graph.

1. The graph G is an empty graph if and only if $\chi(G) = 1$,
2. the graph G is bipartite if and only if $\chi(G) \leq 2$,
3. if G is a forest or an even cycle, then $\chi(G) = 2$,
4. if G is an odd cycle, then $\chi(G) = 3$,
5. if H is a subgraph of G , then $\chi(H) \leq \chi(G)$,
6. if G has clique number $\omega(G)$, then $\omega(G) \leq \chi(G)$,
7. if G has order p , then G is the complete graph K_p if and only if $\chi(G) = p$,
8. if G has connected components G_1, \dots, G_k , then the chromatic number of G is the maximum of the chromatic numbers of its connected components: $\chi(G) = \max(\chi(G_1), \dots, \chi(G_k))$.

In particular, the previous theorem implies that the chromatic number of the Petersen graph is 3 as it contains an odd cycle. Another way to obtain the upper bound for the Petersen graph is to use Brook's Theorem.

Theorem 1.37 (Brooks' Theorem [30]). *Let G be a connected graph, we have $\chi(G) \leq \Delta(G) + 1$. Moreover $\chi(G) = \Delta(G) + 1$ if and only if G is a complete graph or G is an odd cycle.*

We often like to find bounds on the chromatic number of a graph which either depend on other graphs parameters (like Brook's Theorem) or which are only true for particular graphs classes. The most famous theorem in this regard is the Four-Color Theorem.

Theorem 1.38 (Four-Color Theorem [8, 9]). *If G is a planar graph, then $\chi(G) \leq 4$.*

The proof of the Four-Color Theorem is famous as it is the first proof of a major result to be computer assisted. Indeed it relies on the verification of 1834 configurations (this number was later reduced) which is something complicated for a human to do. It uses a process call "discharging" that we will use in Chapter 9. See Figure 1.17 for an example of a 4-coloring of a planar graph.

Note that it is easy to prove that six colors suffices for planar graphs as they are 5-degenerate. It is also not that hard to show that five colors suffices.

As mentioned there is a link between degeneracy and the chromatic number.

Theorem 1.39 (Folklore). *If G is a d -degenerate graph, then $\chi(G) \leq d + 1$.*

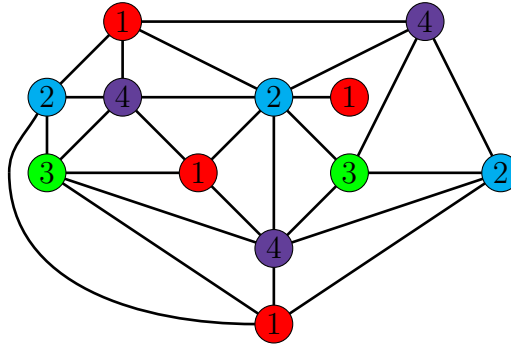


Figure 1.17: A proper vertex 4-coloring of a planar graph.

1.2.2 Homomorphisms

Another point of view on proper vertex coloring is through the concept of homomorphisms.

Definition 1.40 (Homomorphism). A *homomorphism* from G to H is a function φ from $V(G)$ to $V(H)$ such that for every $x, y \in V(G)$, $xy \in E(G)$ implies $\varphi(x)\varphi(y) \in E(H)$. When there is a homomorphism φ from G to H , we note $\varphi : G \rightarrow H$ or simply $G \rightarrow H$.

See Figure 1.18 for an example of homomorphism.

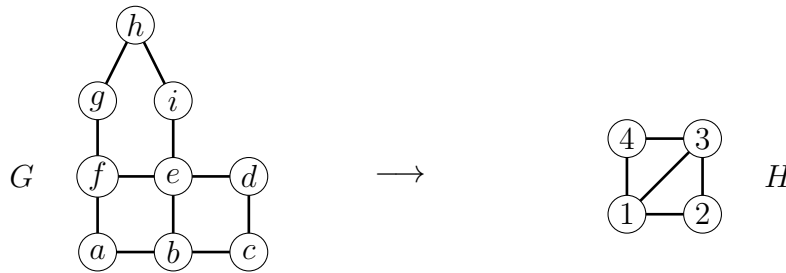


Figure 1.18: Two graphs G and H such that $G \rightarrow H$. One such homomorphism φ can be defined as $\varphi(a) = \varphi(c) = \varphi(g) = \varphi(i) = 1$, $\varphi(b) = 2$, $\varphi(e) = 3$ and $\varphi(d) = \varphi(f) = \varphi(h) = 4$.

Note that one can compose homomorphisms: if $G \rightarrow H$ and $H \rightarrow J$, then $G \rightarrow J$. Also note that a homomorphism from G to H does not need to be surjective, H can have more vertices and edges than the image of G .

Let u and v be two non-adjacent vertices of a graph G . *Identifying* u and v consists in creating the graph H from G , where $V(H) = V(G) \setminus \{u, v\} \cup \{uv\}$ and $E(H)$ is the set $\{wz \in E(G) : w, z \in V(G) \setminus \{u, v\}\} \cup \{(uv, w) : w \in (N_G(u) \cup N_G(v))\}$. In other words, H is the graph obtained by replacing u and v by a new vertex uv which is adjacent to every vertex adjacent to u or v . In particular, if H is the graph obtained from G by identifying u and v , then $G \rightarrow H$ and the corresponding homomorphism is surjective. See Figure 1.19 for an example of identification.

As mentioned before, there is a deep link between the notion of homomorphism and the notion of proper vertex coloring.

Proposition 1.41. *Let G be a graph and k be an integer. The graph G admits a proper k -coloring if and only if $G \rightarrow K_k$.*

The previous proposition implies that one can transform a homomorphism from G into a coloring of G . One way to transform a coloring into a homomorphism is to successively identify vertices of the same color. This implies that we can use these two notions

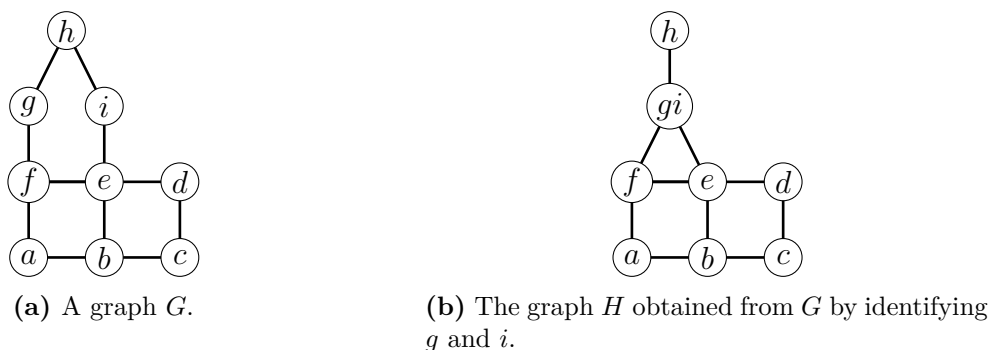


Figure 1.19: Identification of vertices.

interchangeably. See Figure 1.20 for an example of a homomorphism constructed from a coloring.

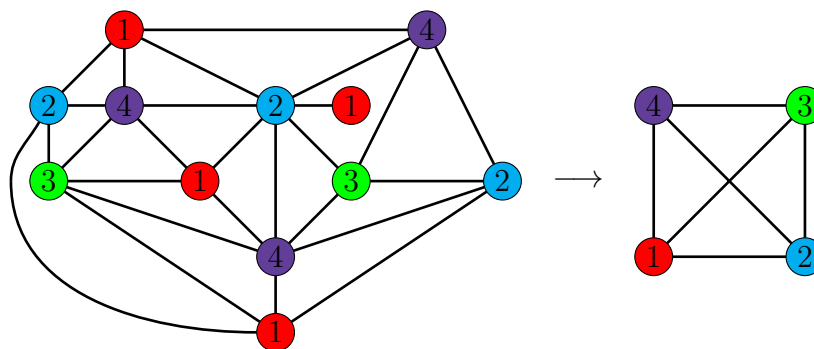


Figure 1.20: The proper vertex 4-coloring of the planar graph in Figure 1.17 transformed into a homomorphism to K_4 .

Definition 1.42 (H -coloring). If φ is a homomorphism from G to H , we say that φ is a H -coloring of G . Indeed, one can see the vertices of H as colors in a vertex coloring. Hence the chromatic number of G can also be defined as the smallest order of a simple graph H such that G admits a H -coloring. As seen in Proposition 1.41, we can always suppose that such a graph H is complete.

The *core* of a graph H is the smallest subgraph C of H such that $H \rightarrow C$. A *core* is a graph which is his own core. The core C of a graph H is unique but one might find multiples copies of C in H . When trying to construct a homomorphism from G to H , it is interesting to replace H by the core of H as G admits a homomorphism to H if and only if it admits a homomorphism to the core of H . For example, the core of a bipartite graph is K_2 . See Figure 1.21 for some examples.

1.2.3 Homomorphisms on more complex graphs

The notion of homomorphism does not limit itself to undirected graphs. In each example below, we present the definition of a homomorphism for different varieties of graphs, in each case, we can derive the notions of core and H -coloring as in the previous section.

First it is possible to consider homomorphisms of multi-graphs.



Figure 1.21: Some graphs and their core represented in bold edges.

Definition 1.43 (Homomorphism of multi-graphs). For two multi-graphs G and H , a *homomorphism* from G to H is a function $\varphi : V(G) \rightarrow V(H)$ such that if $e \in E(G)$ is incident with u and v then there is an edge $e' \in E(H)$ incident with $\varphi(u)$ and $\varphi(v)$. When we want to distinguish the parallel edges, one might require that φ also associate to an edge of G , incident with u and v , an edge of H among all edges incident with $\varphi(u)$ and $\varphi(v)$.

Note that a graph with parallel edges is not a core, hence when trying to decide whether a multi-graph G admits a homomorphism to another multi-graph H , it is better to remove all but one edge of each set of parallel edges in H . Therefore, in this setting, we do not need to distinguish the parallel edges.

For oriented and directed graphs, there is also a notion of homomorphism.

Definition 1.44 (Homomorphism of directed/oriented graphs). For two directed/oriented graphs \vec{G} and \vec{H} , a *homomorphism* from \vec{G} to \vec{H} is a function $\varphi : V(\vec{G}) \rightarrow V(\vec{H})$ such that if \vec{uv} is an arc of \vec{G} then there is an arc $\vec{\varphi(u)\varphi(v)}$ in \vec{H} .

This notion is particularly important for oriented colorings.

Definition 1.45 (Oriented k -coloring). An *oriented k -coloring* of an oriented graph \vec{G} is function φ from $V(\vec{G})$ to $[[k]]$ such that for every arc \vec{uv} , $\varphi(u) \neq \varphi(v)$ and such that no two arcs \vec{uv} and \vec{xy} verify $\varphi(u) = \varphi(y)$ and $\varphi(v) = \varphi(x)$. In other words, an oriented k -coloring is a k -coloring in which all arcs between a vertex colored i and a vertex colored j have the same orientation. The *oriented chromatic number* of \vec{G} , denoted $\chi_o(\vec{G})$, is the smallest k for which \vec{G} admits an oriented k -coloring.

As hinted before, an oriented graph \vec{G} admits an oriented k -coloring if and only if \vec{G} admits a homomorphism to an oriented graph of order at most k .

Finally, we can define the notion of homomorphism for (n, m) -mixed graphs.

Definition 1.46 (Homomorphism of (n, m) -mixed graphs). Let G be a (n, m) -mixed graph and H be a (p, q) -mixed graph. The function $\varphi : V(G) \rightarrow V(H)$ is a *homomorphism* from G to H if and only if:

1. for every edge $uv \in E_i(G)$, $i \leq n$, we have $p \geq i$ and $\varphi(u)\varphi(v) \in E_i(H)$,
2. for every arc $\vec{uv} \in A_j(G)$, $j \leq m$, we have $q \geq j$ and $\vec{\varphi(u)\varphi(v)} \in A_j(H)$.

In this case, we note $G \rightarrow_{(n,m)} H$.

Note that, for this definition, G and H do not have to be the same kind of (n, m) -mixed graphs, this is an abuse of terminology. This notion homomorphism can be specialized for k -edge-colored graphs. For simplicity, we note $G \rightarrow_k H$ instead of $G \rightarrow_{(k,0)} H$.

1.2.4 Edge-coloring

Our second coloring notion is the notion of edge-coloring.

Definition 1.47 (Edge-coloring and chromatic index). A k -edge-coloring of a graph G is a function from $E(G)$ to the set of colors $\llbracket k \rrbracket$. A k -edge-coloring of a graph G is *proper* if no two adjacent edges receive the same color.

The *chromatic index* $\chi'(G)$ of a graph G is the smallest k such that G admits a proper k -edge-coloring. By definition, $\chi'(G) = \chi(L(G))$ where $L(G)$ is the line graph of G .

See Figure 1.22 for an example of a proper 4-edge-coloring of the Petersen graph.

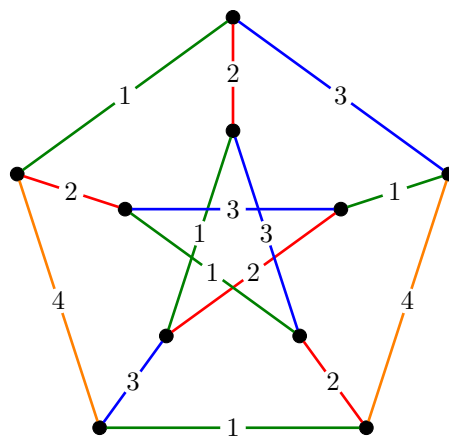


Figure 1.22: A proper 4-edge-coloring of the Petersen graph.

Note that, contrary to vertex coloring, the definition of proper edge-coloring works perfectly for multi-graphs. Also, it is important not to confound a k -edge-colored graph G_1 and a k -edge-coloring of a graph G_2 . In the first case, the edge-coloring is often non-proper and is part of the structure of the graph. In the second case, the edge-coloring is something constructed for the graph G_2 , it is an additional object that complements the graph but is not part of it.

As for the chromatic number, let us see some easy facts about the chromatic index.

Proposition 1.48 (Folklore). *Let G be a graph. The following statements are true.*

1. *If G has maximum degree $\Delta(G)$, then $\chi'(G) \geq \Delta(G)$,*
2. *if G is a forest or an even cycle, then $\chi'(G) = \Delta(G)$,*
3. *if G is an odd cycle, then $\chi'(G) = 3$,*
4. *if H is a subgraph of G , then $\chi'(H) \leq \chi'(G)$,*
5. *if G has connected components G_1, \dots, G_k , then the chromatic index of G is the maximum of the chromatic indices of its connected components:*

$$\chi'(G) = \max(\chi'(G_1), \dots, \chi'(G_k)).$$

In particular, the previous proposition implies that the chromatic index of the Petersen graph is at least 3 as it contains a vertex of degree 3. In fact one can show that three colors are not sufficient in order to properly edge-color the Petersen graph.

In Section 1.2.1, we presented Brooks' Theorem. Note that there is a similar theorem for edge-coloring: Vizing's Theorem.

Theorem 1.49 (Vizing's Theorem [184]). *Let G be a simple connected graph, we have $\chi'(G) \leq \Delta(G) + 1$.*

Note that contrary to Brooks' Theorem, Vizing's Theorem does not characterize the graphs for which $\Delta(G)$ colors suffices, called class I graphs, and the graphs which require $\Delta(G) + 1$ colors, called class II graphs. Since then, no universal criteria has been found to determine whether a graph is class I or class II.

Remark. In the context of distinguishing labellings, an edge-coloring is also called a *labelling* and an edge-color is called a *label*.

1.2.5 List coloring

Often we can convert a notion of coloring into its list version. Let us start with vertex coloring.

Definition 1.50 (List coloring [66]). A *k-list assignment* L for a graph G is a function which associates with each vertex of G a (finite) subset of \mathbb{N} of size at least k . For a list assignment L of G , a (*vertex*) *L-coloring* is a choice function c which maps every vertex $u \in V(G)$ to an element $c(u) \in L(u)$. An *L-coloring* c is *proper* if and only if c is a proper vertex coloring of G . In this case, we say that G is *L-choosable*.

The *choosability* of a graph G , denoted $\text{ch}(G)$, is the smallest $k \in \mathbb{N}$ for which G is *L-choosable* for every *k-list assignment* L of G . A graph of choosability k is called a *k-choosable graph*.

Again note that $\text{ch}(G) \leq |V(G)|$ for every graph G . Also note that a proper *k-coloring* is a proper *L-coloring* of G where $L(u) = \llbracket k \rrbracket$ for every $u \in V(G)$, hence $\chi(G) \leq \text{ch}(G)$. Note however that the gap between $\chi(G)$ and $\text{ch}(G)$ can be arbitrarily large.

Theorem 1.51 ([66]). *For every $k \in \mathbb{N}$, there exists a bipartite graph G with choosability at least k .*

The notion of edge coloring also has its list version.

Definition 1.52 (List edge-coloring [66]). A *k-list edge-assignment* L for a graph G is a function which associates with each edge of G a (finite) subset of \mathbb{N} of size at least k . For a list edge assignment L of G , an *L-edge-coloring* is a choice function c which maps every edge $e \in E(G)$ to an element $c(e) \in L(e)$. An *L-edge-coloring* c is *proper* if and only if c is a proper edge-coloring of G . In this case, we say that G is *L-edge-choosable*.

The *edge-choosability* of a graph G , denoted $\text{ch}'(G)$, is the smallest $k \in \mathbb{N}$ for which G is *L-edge-choosable* for every *k-list edge-assignment* L of G .

Note that a proper *k-edge-coloring* is a proper *L-edge-coloring* of G where $L(e) = \llbracket k \rrbracket$ for every $e \in E(G)$, hence $\chi'(G) \leq \text{ch}'(G)$. Contrary to the vertex coloring case, it is conjectured that $\chi'(G) = \text{ch}'(G)$ for every graph G .

Remark. In the context of distinguishing labellings, an L -edge-coloring is also called an L -labelling.

Let us see why the concept of list coloring is useful. We only consider here the vertex coloring case but the same arguments apply for edge-coloring. Suppose that we want to find a proper k -coloring of a graph G . Let H be a “small” subgraph of G such that there is a proper vertex k -coloring c_1 of $G - V(H)$ (obtained by induction, for example).

To color G , we want to extend the coloring c_1 of $G - V(H)$ to the vertices of H . For each vertex u of $V(H)$, we create a list $L(u)$ containing the colors among $\llbracket k \rrbracket$ which are not already assigned to a neighbor of u in $G - V(H)$ by c_1 . If H admits a proper L -coloring c_2 , then the coloring c of G , defined by $c(u) = c_1(u)$ if $u \in V(G) \setminus V(H)$ and $c(u) = c_2(u)$ if $u \in V(H)$, is a proper vertex k -coloring of G .

Note that this does not hold if we take any arbitrary proper k -coloring of H . Also note that the difficult parts in this process are finding the subgraph H and the L -coloring c_2 . We present in Section 1.4.4 a way to find this L -coloring by using an algebraic method: the Combinatorial Nullstellensatz.

1.3 NP-completeness and FPT

In this section, we introduce various complexity notions that are used throughout this thesis. We start with a quick reminder of general complexity notions in section 1.3.1 and a quick overview of different ways to represent graphs in algorithms in Section 1.3.2. Section 1.3.3 presents the concept of NP-complete problem while Section 1.3.4 focuses on parameterized complexity. We conclude this section with a list of additional complexity problems in Section 1.3.5.

Since the theory of calculability and Turing machines are not the focus of this thesis, we do not get into the rigorous formalism of the field here. Nonetheless, we refer the reader to [53, 160, 173] for more details on the topics covered in this section.

1.3.1 Basics of Complexity Theory

An *algorithm* \mathcal{A} is a sequence of elementary operations (see RAM model [160, section 2.6]) which can be performed on an input x respecting a given format (*e.g.* the input is an integer, a graph ...).

The *time* $t_{\mathcal{A}}(x)$ of an algorithm \mathcal{A} on input x is the total number of elementary operations performed by \mathcal{A} on input x before \mathcal{A} stops. For every non-negative integer n , the *time complexity* $t_{\mathcal{A}}(n)$ of an algorithm \mathcal{A} is the maximum of $t_{\mathcal{A}}(x)$ over all inputs x of size at most n .

The *space* $s_{\mathcal{A}}(x)$ of an algorithm \mathcal{A} on input x is the maximum number of memory locations used simultaneously by \mathcal{A} on input x before \mathcal{A} stops. For every non-negative integer n , the *space complexity* $s_{\mathcal{A}}(n)$ of an algorithm \mathcal{A} is the maximum of $s_{\mathcal{A}}(x)$ over all inputs x of size at most n .

While we mainly focus on time complexity from now on, note that analogous notions exist for space complexity.

We characterize the complexity of an algorithm \mathcal{A} by the growth of the function $t_{\mathcal{A}}(n)$: an algorithm is *linear* (resp. *polynomial*, resp. *exponential* ...) if $t_{\mathcal{A}}(n) = O(n)$ (resp. $t_{\mathcal{A}}(n) = O(n^c)$ for some constant c , resp. $t_{\mathcal{A}}(n) = 2^{O(n^c)}$ for some positive constant c ...).

A *problem* \mathcal{P} consists in solving a given problematic given an input. A problem is a *decision problem* if the problematic is a yes/no question. An instance \mathcal{I} of \mathcal{P} is a *positive instance* of \mathcal{P} if and only if the answer to the problematic of \mathcal{P} is yes. An algorithm \mathcal{A} solves a problem \mathcal{P} if for every input x of the problem, \mathcal{A} outputs $\mathcal{A}(x)$ such that $\mathcal{A}(x)$ is the answer to the question of the problem on input x . An example of a problem is:

ARRAY SEARCH

Input: An array A , its size n and an element x .

Output: The index of x in A if x is in A and -1 otherwise.

A problem \mathcal{P} is in \mathbf{P} if there exists a polynomial algorithm solving \mathcal{P} . Intuitively, \mathbf{P} is the class of problems that can be solved in “reasonable time”. A goal of the Complexity Theory is to determine which problems belong to \mathbf{P} and which problems do not. For example, the problem ARRAY SEARCH is in \mathbf{P} as it suffices to go sequentially through the array to find the solution.

Another important class of problems is the class \mathbf{NP} , for Non-deterministic Polynomial. It can be informally defined as the set of decision problems for which we can verify, in polynomial time, whether some given candidate solution x is really a solution of the problem. Intuitively, \mathbf{NP} is the class of problems for which we can verify if an input is a solution in “reasonable time”. In particular, $\mathbf{P} \subseteq \mathbf{NP}$. For more precise definitions (in terms of non-deterministic Turing machines), we refer the reader to [160, 173]. For example, the following problem is in \mathbf{NP} .

3-COLORING

Input: A graph G .

Question: Does the graph G admit a proper 3-coloring?

Indeed, even if it is not simple to construct a proper 3-coloring of a graph G , if we are given a 3-coloring of G , it is easy to verify that this coloring is proper.

1.3.2 Algorithmic representations of graphs

In this section, we see how to represent graphs in algorithms. Let G be a graph with $V(G) = \{v_1, \dots, v_n\}$. We give an example of such a graph in Figure 1.23 and its two representations in Figure 1.24 and Figure 1.25.

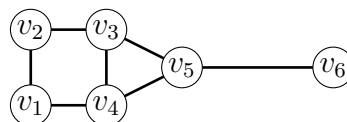


Figure 1.23: A graph G on 6 vertices.

The first method to represent G is by an *adjacency matrix* $M \in \mathcal{M}_n(\{0, 1\})$, *i.e.* a matrix of size n by n such that m_{ij} , the coefficient of line i and column j , is 0 if $v_i v_j \notin E(G)$ and 1 otherwise. For this data structure, it takes constant time to test whether two vertices are adjacent, however, it takes $O(n)$ time to recover all the neighbors of a given vertex. This data structure is more adapted for dense graphs where the number of edges m is $\Theta(n^2)$. Finally, note that the adjacency matrix of an undirected graph is symmetric.

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1.24: The adjacency matrix of the graph G .

The other method to represent G is by an *adjacency list*, i.e. an array A of n lists where the cell $A[i]$ contains the list of neighbors of the vertex v_i . For this data structure, it takes $O(\Delta(G))$ time to test whether two vertices are adjacent, however, it takes $O(d(u))$ time to recover all the neighbors of a given vertex u . Note that if the lists are implemented as sorted arrays, then it actually takes $\log(\Delta(G))$ time to find if two vertices are adjacent using dichotomy. This method is more adapted for sparse graphs where the number of edges m is $O(n)$, like for planar graphs.

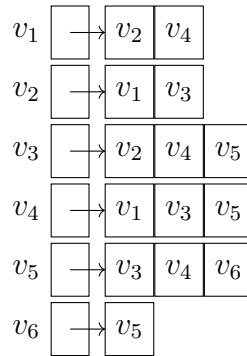


Figure 1.25: The adjacency list of the graph G .

The choice of the representation depends on which operations on graphs are more important in the algorithm. If it is more important to test the adjacency, then we prefer using adjacency matrices. If it is more important to traverse the graph (and thus to get the list of neighbors of a vertex), then we prefer using adjacency lists.

Note that there exist more data structures which represent graphs. For more complex problems, it is often efficient to add more information in the data structure, like representing faces for planar graphs. One might even find useful to have both the adjacency matrix and the adjacency list of a graph.

1.3.3 NP-completeness

One of the major questions in Complexity Theory is to determine whether $P = NP$, or not; the problem might even be undecidable. This problem is one of the Millennium Prize Problems stated by the Clay Mathematics Institute in 2000 and has gathered a lot of attention through the years.

One approach used to attack the question is to try to show that the “hardest” problems in NP can be solved in polynomial time. While this has yet to be achieved, the developed notions of reductions and NP-completeness are powerful tools to show that it is very unlikely that some problems admit polynomial time algorithms.

Definition 1.53 (Polynomial reduction). A decision problem \mathcal{P} *reduces* (in polynomial time) to a decision problem \mathcal{P}' if and only if there exists a computable polynomial function f (i.e. an algorithm) such that for every instance \mathcal{I} of \mathcal{P} , \mathcal{I} is a positive instance of \mathcal{P} if and only if $f(\mathcal{I})$ is a positive instance of \mathcal{P}' .

Definition 1.54 (NP-hardness, NP-completeness). A problem \mathcal{P} is *NP-hard* if any problem \mathcal{P}' in NP reduces to \mathcal{P} . A problem is *NP-complete* if it is NP-hard and belongs to the class NP.

Intuitively, NP-complete problems are the hardest problems in the NP complexity class. The most known of these is the SAT-CNF problem. In order to present this problem, we need to talk first about Boolean formulas.

Definition 1.55 (Boolean formulas). The set of *Boolean formulas* \mathcal{BF} is constructed as follows:

1. \mathcal{BF} contains all variables x_1, x_2, \dots ,
2. if φ is a Boolean formula, \mathcal{BF} contains $\neg\varphi$, the negation of φ which is true if and only if φ is false,
3. if φ_1 and φ_2 are two Boolean formulas, \mathcal{BF} contains $\varphi_1 \wedge \varphi_2$, the conjunction of φ_1 and φ_2 which is true if and only if φ_1 and φ_2 are both true,
4. if φ_1 and φ_2 are two Boolean formulas, \mathcal{BF} contains $\varphi_1 \vee \varphi_2$, the disjunction of φ_1 and φ_2 which is false if and only if φ_1 and φ_2 are both false.

To simplify notation, we often note \bar{x}_i for $\neg x_i$ when x_i is a variable. A *literal* is either a variable x_i or its negation \bar{x}_i . A *clause* C is a disjunction of literals, i.e. $C = (\ell_1 \vee \ell_2 \vee \dots \vee \ell_k)$ where each ℓ_i is a literal. The *size* of a clause C is the number of literals in the clause. A Boolean formula φ is in *conjunctive normal form (CNF)* if it is a conjunction of clauses, i.e. $\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ where each C_i is a clause. For concision, we use the term CNF formula to refer to a Boolean formula in conjunctive normal form. A *variable assignment* is a function from the set of variables to the set of *truth values* {true, false}. A variable assignment *satisfies* a CNF formula if and only if there is at least one true literal in each clause.

We can now define the SAT-CNF problem.

SAT-CNF

Input: A Boolean formula φ in conjunctive normal form with n variables and m clauses.

Question: Is there a variable assignment satisfying φ ?

This problem was the first to be shown to be NP-complete by Cook [50] and Levin [139] independently.

Theorem 1.56 (Cook-Levin [50, 139]). *The problem SAT-CNF is NP-complete.*

We present two other famous problems:

2-SAT

Input: A Boolean formula φ in conjunctive normal form with n variables and m clauses where each clause has size at most 2.

Question: Is there a variable assignment satisfying φ ?

3-SAT

Input: A Boolean formula φ in conjunctive normal form with n variables and m clauses where each clause has size at most 3.

Question: Is there a variable assignment satisfying φ ?

One can easily generalize the previous problems to k -SAT. It is known since 1967, with the work of Krom, that 2-SAT is in P. The key idea is to rewrite a 2-SAT formula as a sequence of implications. However no such algorithm has been found for 3-SAT, in fact 3-SAT is one of the 21 NP-complete problems presented by Karp in his 1972 paper [126].

Theorem 1.57 ([126]). *The problem 3-SAT is NP-complete.*

The proof is done by reduction (see Definition 1.53). We first present the general methodology of a proof by reduction before using the proof of Theorem 1.57 as an example of this type of proofs, that we will use in Chapter 3 and Chapter 8.

Proof of NP-completeness by reduction. Let \mathcal{P} be our problem. The first step of such a proof is to show that \mathcal{P} is in NP. This is generally quite easy as it suffices to show that one can verify that a proposed solution is indeed a solution of \mathcal{P} . This usually boils down to verifying all the required constraints on the solution.

The second step is the actual reduction. Let \mathcal{P}' be an NP-hard problem, we want to show that \mathcal{P}' reduces to \mathcal{P} . For this, take an arbitrary instance \mathcal{I}' of \mathcal{P}' . We want to create an instance \mathcal{I} of \mathcal{P} which “encodes” the instance \mathcal{I}' of \mathcal{P}' . The goal is to show that \mathcal{I}' is a solution of \mathcal{P}' if and only if \mathcal{I} is a solution of \mathcal{P} .

Let us see an example of this type of proof.

Proof of Theorem 1.57. First, as SAT-CNF is in NP, and since an instance of 3-SAT is also an instance of SAT-CNF, 3-SAT is in NP.

We want to reduce SAT-CNF to 3-SAT. Let φ be a CNF Boolean formula instance of SAT-CNF on n variables x_1, \dots, x_n and m clauses C_1, \dots, C_m . Fix $i \in \{1, \dots, m\}$ and suppose that $C_i = (\ell_1^i \vee \dots \vee \ell_{k_i}^i)$ where each ℓ_j^i is a literal. The goal is to create a new set of clauses for 3-SAT that will mimic the clause C_i .

Let C^i be the following set of clauses:

$$(\ell_1^i \vee \ell_2^i \vee x_1^i) \wedge (\overline{x_1^i} \vee \ell_3^i \vee x_2^i) \wedge (\overline{x_2^i} \vee \ell_4^i \vee x_3^i) \wedge \dots \wedge (\overline{x_{k_i-3}^i} \vee \ell_{k_i-1}^i \vee \ell_{k_i}^i)$$

where the x_j^i are new variables. Clearly, C^i is satisfiable if and only if at least one of the literals ℓ_j^i is true. Hence $\bigwedge_{i=1}^m C_i$ is satisfiable if and only if $\bigwedge_{i=1}^m C^i$ is satisfiable. This concludes the proof. \square

From the previous proof, one can see that we used clauses of size 3 in order to represent a clause of arbitrary size. Sometimes, the reductions are more convoluted but the principle always stays the same. In Chapter 3 and Chapter 8, we perform such reductions on several graph problems to show their NP-completeness.

Showing the NP-completeness of a problem indicates that it is unlikely that the problem can be solved in polynomial time, but, depending on the reduction, the size of the constructed instance might grow linearly, quadratically, or even worse, with the size of the original instance. This fact is unfortunately lost in the statement of the theorem. The following hypothesis allows us to be more precise.

Definition 1.58 (Exponential Time Hypothesis (ETH) [145]). The *Exponential Time Hypothesis* (ETH) postulates that 3-SAT cannot be solved in time $2^{o(n)}(n+m)^c$, where n and m are the input's number of variables and clauses, and c is any integer. In particular, this implies that 3-SAT cannot be solved in time $2^{o(n+m)}$ (see [53]).

Suppose that 3-SAT reduces to a problem \mathcal{P} such that for an instance of 3-SAT with n variables and m clauses, the constructed instance of \mathcal{P} has size $O((n+m)^2)$. Then, assuming ETH, we can show that there is no algorithm solving \mathcal{P} in time $2^{o(\sqrt{s})}s^c$, for any constant integer c , where s is the size of the instance of \mathcal{P} . Indeed, if such an algorithm existed we could solve 3-SAT in time $2^{o(n+m)}(n+m)^{2c} = 2^{o(n+m)}$, contradicting ETH. Even if the ETH was to be false, this type of result shows that \mathcal{P} seems “easier” to solve than 3-SAT is.

This highlights how ETH can help us to be more precise than just saying “ \mathcal{P} is NP-complete”. Here the function in the o notation indicates how powerful the problem is compared to 3-SAT.

1.3.4 Parameterized complexity

A way to construct efficient algorithms for NP-complete problems is to lower our time complexity expectations. Usually, we consider an algorithm efficient if it is polynomial. By adding an extra parameter k to the problem, we would like to keep the polynomiality in n , the size of the instance, while “pushing” the hardness of the problem on the parameter k . For more details on parameterized complexity, we refer the reader to [53].

Definition 1.59 (Parameter). A *parameter* is a number k which is either given by the problem, *e.g.* an integer when the problem consists in determining whether there is a solution smaller than k or not, or given as a function of the input, *e.g.* the treewidth of the input graph.

For example, VERTEX COVER is one of the 21 NP-complete problems of Karp [126]. Let us consider a version parameterized by the solution size.

VERTEX COVER

Parameter: k .

Input: A graph G and an integer k .

Question: Is there a set S of at most k vertices of G , such that, for each edge uv of G , at least one of u or v belongs to S ?

Definition 1.60 (Fixed Parameter Tractable). A problem \mathcal{P} of size n and parameterized by k is *Fixed Parameter Tractable* (FPT) if and only if there exists an algorithm solving \mathcal{P} in time $f(k) \cdot O(n^c)$ where c is a fixed integer and f is any computable function.

Theorem 1.61 (Folklore). *The problem VERTEX COVER is FPT and can be solved in time $2^{O(k)}n^2$.*

The previous theorem does not give a polynomial complexity but the non-polynomial part of the complexity part does not depend on the size of the problem n . It is of course possible to improve the previous theorem. The best known bound at the time of writing this thesis is $O(1.2738^k + kn)$ by Chen, Kanj and Xia [44].

When working with the treewidth as a parameter, we can also use Courcelle's theorem. This theorem relies on the notion of monadic second-order logic of graphs which we do not define here (for more details see [51]).

Theorem 1.62 (Courcelle’s Theorem [51]). *Every graph property definable in the monadic second-order logic of graphs can be decided in linear time on graphs of bounded treewidth, i.e. for a graph G of order n the problem can be decided in time $f(\text{tw}(G))n$.*

Note that Courcelle’s theorem proves that the problem of deciding whether such a property holds or not for a given graph is FPT but the generality of this theorem imposes that the function f is far from optimal (it is a tower of exponentials). This is why it is often better to construct the algorithm directly to obtain a better function of k .

It is not always possible to find an FPT algorithm for a given problem. Let XP be the class of problems which can be solved in time $O(n^{f(k)})$ where f is a computable function. Note that FPT is included in XP.

Among the problems in XP, a particular class of problems is the class W[i] where i is an integer. We do not give the exact definition of these classes here but one can find it in [53, Chapter 13]. By definition $W[0] = \text{FPT}$ and $W[i] \subseteq W[j]$ when $i \leq j$. By analogy with NP, we say that a problem \mathcal{P} is W[i]-hard if every problem in W[i] reduces to \mathcal{P} in FPT-time.

Definition 1.63 (Parameterized reduction). A parameterized decision problem \mathcal{P} reduces (in FPT time) to a parameterized decision problem \mathcal{P}' if and only if for every instance \mathcal{I} of \mathcal{P} , \mathcal{I} is a positive instance of \mathcal{P} with parameter k if and only if $f(\mathcal{I}, k)$ is a positive instance of \mathcal{P}' with parameter $g(k)$ where g is a polynomial computable function and f is a computable function such that the size of $f(\mathcal{I}, k)$ is of the form $O(h(k) |\mathcal{I}|^c)$ where h is a computable function and c is a fixed integer.

Here we are only interested in the class W[1]. The class W[1] can be seen as the class of parameterized problems to which the problem INDEPENDENT SET, parameterized by the solution size, reduces to in FPT-time (see [57]). Showing that a parameterized problem is W[1]-complete implies that it is unlikely that there exists an FPT algorithm solving it. Here is an example of such a problem.

MULTICOLORED INDEPENDENT SET

Parameter: k .

Input: A graph G , an integer k and a partition of $V(G)$ into k sets V_1, \dots, V_k .

Question: Is there a set S of exactly k vertices of G , such that each V_i contains exactly one element of S , and S an independent set of G ?

Theorem 1.64 ([162]). *MULTICOLORED INDEPENDENT SET is W[1]-complete.*

Hence, the previous theorem implies that it is highly unlikely that MULTICOLORED INDEPENDENT SET admits an FPT algorithm.

Again using the ETH allows us to be more precise in the statement thanks to this result.

Theorem 1.65 ([43]). *Assuming ETH, MULTICOLORED INDEPENDENT SET cannot be solved in time $f(k) |V(G)|^c$ where f is a computable function, and c is a fixed integer.*

Theorem 1.66 (Consequence of [126], see [53]). *Assuming ETH, VERTEX COVER, parameterized by the solution size k , cannot be solved in time $2^{o(k)} |V(G)|^c$ where c is a fixed integer.*

1.3.5 Some problems used in reductions

In this section, we list a number of problems and present their complexity. These problems are used in Chapter 3 and Chapter 8.

ODD CYCLE TRANSVERSAL **Parameter:** k .

Input: A graph G , an integer k .

Question: Is there a set of k vertices of G that can be deleted from G so that the resulting graph is bipartite?

The NP-completeness of ODD CYCLE TRANSVERSAL follows from a result of Yannakakis [206]. Moreover, ODD CYCLE TRANSVERSAL is FPT (see [166, 127]).

EDGE BIPARTIZATION **Parameter:** k .

Input: A graph G , an integer k .

Question: Is there a set of k edges of G that can be deleted from G so that the resulting graph is bipartite?

EDGE BIPARTIZATION is NP-complete (see [83]) and even FPT (see [88, 163]).

VARIABLE DELETION ALMOST 2-SAT **Parameter:** k .

Input: A 2-CNF Boolean formula F and an integer k .

Question: Is there a set of k variables that can be deleted from F (together with the clauses containing them) so that the resulting formula is satisfiable?

Even if 2-SAT is in P, VARIABLE DELETION ALMOST 2-SAT is NP-complete and even FPT (see [53, Chapter 3.4]).

Example of proof of NP-completeness. First, note that the problem is in NP. The NP-hardness follows from a reduction to VERTEX COVER. Let G be a graph and k be a integer number. Introduce for each vertex v of G the variable x_v . For each edge uv introduce the set of clauses $C_{uv}: (x_v \vee x_u) \wedge (x_v \vee \bar{x}_u) \wedge (\bar{x}_v \vee x_u) \wedge (\bar{x}_v \vee \bar{x}_u)$. We obtain a CNF formula F .

If the graph G has a vertex cover S of size at most k , then we remove the variables corresponding to the vertices of S . This removes all the clauses of F and thus we obtain CNF formula which is satisfiable. Conversely, if there is a set S of at most k variables whose removal from F creates a satisfiable formula, then we remove the vertices corresponding to the variables of S in G . After removal, if some edge uv remains in G , then C_{uv} remains in F after the removal of S . This is a contradiction as C_{uv} is a non-satisfiable set of clauses. \square

CLAUSE DELETION ALMOST 2-SAT **Parameter:** k .

Input: A 2-CNF Boolean formula F and an integer k .

Question: Is there a set of k clauses that can be deleted from F so that the resulting formula is satisfiable?

CLAUSE DELETION ALMOST 2-SAT is NP-complete (see [53, Exercice 3.21]) and even FPT [165].

k -EDGE-COLORING

Input: A graph G with maximum degree k .

Question: Does G admit a proper k -edge-coloring?

The problem k -EDGE-COLORING was proved to be NP-Complete even for regular graphs in [106] (for the case $k = 3$) and [140] (for the general case).

PLANAR VERTEX 3-COLORING**Input:** A planar graph G with maximum degree 4.**Question:** Does G admit a proper vertex 3-coloring?

The problem PLANAR VERTEX 3-COLORING was proved to be NP-Complete in [83].

1.4 A little bit of algebra

In this section, we recall some classical algebraic notions. While this section is not essential in order to understand this thesis, it eases the understanding of Chapter 5. We also introduce here the Combinatorial Nullstellensatz (Theorem 1.77) which is used in Chapter 7 and Chapter 10. For more details, we refer the reader to [203] (in French) or any bachelor level textbook on algebra.

1.4.1 Well-founded ordering and two classical proof methods in Graph Theory

A number of proofs in this thesis are done either by induction or by minimal counter-example. These two types of proofs are extremely similar and rely on well-founded orders on graphs. We first recall some definitions.

Definition 1.67 (Order). An *order* relation R over a set X is a subset of X^2 , the set of ordered pairs of elements of X , verifying:

1. R is irreflexive (*i.e.* $\forall x \in X, (x, x) \notin R$),
2. R is transitive (*i.e.* $\forall x, y, z \in X, (x, y) \in R$ and $(y, z) \in R$ implies $(x, z) \in R$), and
3. R is antisymmetric (*i.e.* $\forall x, y \in X, (x, y) \in R$ implies $(y, x) \notin R$).

We often note xRy for $(x, y) \in R$. A classical example of this notation is $x < y$ where x and y are integers and $<$ is the order on the natural numbers.

Definition 1.68 (Well-founded order). A *well-founded order* R is an order over a set X for which there is no sequence $(u_i)_{i \in \mathbb{N}}$ such that $(u_{i+1}, u_i) \in R$ for all $i \in \mathbb{N}$, *i.e.* there is no infinitely decreasing sequence for R .

For example, the order $<$ is well-founded on the set of natural numbers but is not well-founded on the set of relative numbers. Let us now present the two proof methods.

Proof by induction. Let \mathcal{P} be a predicate on a set X (for us it would be a set of graphs), and let \prec be a well-founded order on X . To prove $\mathcal{P}(x)$ for every $x \in X$, it suffices to prove that for every $x \in X$, if $\mathcal{P}(y)$ holds for every $y \in X$ with $y \prec x$, then $\mathcal{P}(x)$ holds.

Proof by minimal counter-example. Let \mathcal{P} be a predicate on a set X (for us it would be a set of graphs), and let \prec be a well-founded order on X . Let $A = \{x \in X, \neg \mathcal{P}(x)\}$.

If A is empty, then $\mathcal{P}(x)$ holds for every $x \in X$. Otherwise, by contradiction, consider x_m a minimal element of A . Then by definition, $\mathcal{P}(y)$ holds for every $y \in X$, $y \prec x_m$. In this type of proof, we want to find a contradiction with the existence of x_m .

A proof by minimal counter-example can generally be rewritten as a proof by induction but it is sometimes easier to write it as a minimal counter-example proof. Finally recall that, when coloring graphs, if each part of a proof by induction is constructive, then we can derive an algorithm from the proof to construct the coloring. These algorithms are generally polynomial.

Let us now see how to construct well-founded orders on the set of graphs. One way to achieve this goal is to consider the proper subgraph relation or, the weaker proper induced subgraph relation. Even if these relations are natural, they do not offer much freedom on which “smaller graphs” can be considered.

Another way to construct such orders is to reduce to the order on the natural numbers. Let f be a function from the set of graphs to the set of natural numbers, and for all graphs G and H , define $G \prec_f H$ if and only if $f(G) < f(H)$. It is easy to verify that \prec_f is a well-founded order. Some classical examples are when $f_v(G) = |V(G)|$ or $f_e(G) = |E(G)|$.

Finally we can combine multiple functions from the previous approach with each other. Let f_1, f_2, \dots, f_k be k functions from the set of graphs to the natural numbers. The *lexicographic order* \prec is a well-founded order where \prec is defined by $G \prec H$ if and only if there exists $i \in \{1, \dots, k\}$ such that $f_i(G) < f_i(H)$ and for all $j < i$, $f_j(G) = f_j(H)$. We often note the lexicographic order by listing the functions it is composed of, for example $(|V(G)|, |E(G)|)$ is the lexicographic order created from the functions f_v and f_e of the previous paragraph.

More generally, if $\prec_1, \prec_2, \dots, \prec_k$ are k well-founded orders on the set of graphs, then the *lexicographic order* $(\prec_1, \prec_2, \dots, \prec_k)$, noted \prec for short, is a well-founded order. Here \prec is defined by $G \prec H$ if and only if there exists $i \in \{1, \dots, k\}$ such that $G \prec_i H$ and for all $j < i$, neither $G \prec_j H$ nor $H \prec_j G$.

1.4.2 Equivalence and quotient

A notion that is underlying in the theory of signed graphs (see Part I) is the notion of equivalence. We recall some definitions here.

Definition 1.69 (Equivalence relation). An *equivalence relation* R over a set X is a subset of X^2 , the set of ordered pairs of elements of X , verifying:

1. R is reflexive (*i.e.* $\forall x \in X, (x, x) \in R$),
2. R is transitive (*i.e.* $\forall x, y, z \in X, (x, y) \in R$ and $(y, z) \in R$ implies $(x, z) \in R$), and
3. R is symmetric (*i.e.* $\forall x, y \in X, (x, y) \in R$ implies $(y, x) \in R$).

We often note xRy for $(x, y) \in R$. An example of this notation is $(a, b) =_{\mathbb{Q}} (c, d)$ where a and c are relative integers, b and d are positive integers, and $=_{\mathbb{Q}}$ is equality over the rational numbers defined as $(a, b) =_{\mathbb{Q}} (c, d)$ if and only if $ad = bc$ (*i.e.* $\frac{a}{b} = \frac{c}{d}$ if only if $ad = bc$).

Definition 1.70 (Equivalence class, quotient set). The *equivalence class* of an element $x \in X$ for the relation R , denoted \bar{x} , is the set $\{y \in X : xRy\}$. The *quotient set* of X for the relation R , denoted X/R , is the set $\{\bar{x} : x \in X\}$.

Note that the \bar{x} notation is also used in the context of Boolean formulas, the signification of the notation should always be clear from the context.

Considering a property “up to equivalence” is the same as considering this property in the quotient set. Therefore, a graph can be seen as an element of the quotient of the set of pairs $(V(G), E(G))$ by the isomorphism equivalence relation. In Chapter 5 Section 5.4, we use some notion of quotient on graphs which is defined using the notion of quotient set.

1.4.3 Algebraic structures

Let us start by recalling some algebraic structures.

Definition 1.71 (Monoid). A *monoid* $(G, *)$ is an ordered pair where G is a set and $*$: $G \times G \rightarrow G$ is a binary operation, verifying the following:

1. $*$ is associative (*i.e.* $\forall x, y, z \in G, (x * y) * z = x * (y * z)$),
2. $(G, *)$ has a neutral element e (*i.e.* $\forall x \in G, e * x = x * e = x$).

A monoid is *commutative* if and only if for every $x, y \in G, x * y = y * x$. A commutative monoid is also called an *abelian monoid*.

Definition 1.72 (Group). A *group* $(G, *)$ is a monoid where each element has an inverse (*i.e.* $\forall x \in G, \exists y \in G, x * y = y * x = e$). We generally write x^{-1} for the inverse of x . When using the $+$ binary operation, we prefer to write $-x$ for the inverse of x .

A group $(G, *)$ is *commutative* if and only if $(G, *)$ is commutative as a monoid. A commutative group is also called an *abelian group*.

To simplify notation, we often note G for the group $(G, *)$. If H is a subgroup of G (*i.e.* $H \subseteq G$ and $(H, *)$ is a group), then the *quotient group* G/H is the group $(G/\sim, *)$ where $x \sim y$ if and only if $x * y^{-1} \in H$ and where the $*$ operation, in the quotient, verifies $\overline{x * y} = \overline{x} * \overline{y}$.

Definition 1.73 (Semiring). A *semiring* $(A, +, \times)$ is a triplet where A is a set and $+$ and \times are two binary operations, verifying the following:

1. $(A, +)$ is a commutative monoid with neutral element 0,
2. (A, \times) is a monoid with neutral element 1,
3. \times is distributive over $+$ (*i.e.* $\forall x, y, z \in A, x \times (y + z) = x \times y + x \times z$ and $(y + z) \times x = y \times x + z \times x$),
4. 0 is an absorbing element for \times (*i.e.* $\forall x \in A, 0 \times x = x \times 0 = 0$).

A semiring is *commutative* if and only if (A, \times) is a commutative monoid. A semiring has the *cancellation property* if and only if $\forall x, y, z \in A, (x \times y = x \times z \text{ implies } y = z)$ and $(y \times x = z \times x \text{ implies } y = z)$. A commutative semiring is also called an *abelian semiring*.

Definition 1.74 (Ring). A *ring* $(A, +, \times)$ is a semiring where $(A, +)$ is a group. A ring is *commutative* if and only if (A, \times) is a commutative monoid. A ring has the *cancellation property* if and only if it has the cancellation property as a semiring. A commutative ring is also called an *abelian ring*.

Definition 1.75 (Field). A *field* $(\mathbb{F}, +, \times)$ is a ring where (\mathbb{F}, \times) is an abelian group.

In this thesis, we only consider commutative fields. Note that some authors consider fields to be non-commutative. Note also that a field has the cancellation property.

Recall that a *polynomial* is a finite sum of monomials and a *monomial* is the product of a coefficient and a finite list of indeterminates with possibly some repetitions.

Example 1.76. Some examples in mathematics:

1. The structure $(\mathbb{N}, +)$, where \mathbb{N} is the set of natural numbers, is an abelian monoid.
2. The structure $(\mathbb{Z}, +)$, where \mathbb{Z} is the set of relative numbers, is an abelian group.
3. The structure $(\mathbb{N}, +, \times)$ is an abelian semiring.
4. The structure $(\mathbb{N}[(X_i)_{i \in \mathbb{N}}], +, \times)$, where $\mathbb{N}[(X_i)_{i \in \mathbb{N}}]$ is the set of polynomials with coefficients in \mathbb{N} over the indeterminates $(X_i)_{i \in \mathbb{N}}$, is an abelian semiring.
5. The structure $(\mathbb{Z}, +, \times)$ is an abelian ring.
6. The structures $(\mathbb{Q}, +, \times)$ and $(\mathbb{R}, +, \times)$, where \mathbb{Q} is the set of rational numbers, and \mathbb{R} is the set of real numbers, are fields.
7. The structure $(\{-1, 1\}, \times)$ is an abelian group.
8. The structure $(\mathcal{M}_n(\mathbb{R}), +, \times)$, where $\mathcal{M}_n(\mathbb{R})$ is the set of $n \times n$ matrices over the set of reals, is a non-commutative ring.
9. The structure $(\mathbb{F}[X], +, \times)$, where $\mathbb{F}[X]$ is the set of polynomials with coefficients in the field \mathbb{F} , is an abelian ring.
10. The structure $\mathbb{F}_p = (\mathbb{Z}/p\mathbb{Z}, +, \times)$ is a finite field.

Other examples in graphs:

11. The structure (\mathfrak{G}, \uplus) , where \mathfrak{G} is the set of graphs and \uplus is the disjoint union operation (*i.e.* $G_1 \uplus G_2$ is the graph containing two disjoint copies of G_1 and G_2), is an abelian monoid.
12. The structure $(\mathfrak{G}, \uplus, *)$, where $*$ is either the Cartesian product of graphs \square (see [169]), the tensor product of graphs \times (see [192] volume 2 p.384), the strong product of graphs \boxtimes (see [169]) or the lexicographical product of graphs \cdot (see [96]), is a semiring. You can find more discussion about these products in Chapter 5.

1.4.4 Combinatorial Nullstellensatz

Let us conclude this section by presenting the Combinatorial Nullstellensatz. Take a graph G , an induced subgraph H of G and a k -vertex coloring c of H . One might want to extend c to all vertices of G but this might be very complicated to do manually. Let us see how to use the Combinatorial Nullstellensatz.

Theorem 1.77 (Combinatorial Nullstellensatz [4]). *Let \mathbb{F} be an arbitrary field, and let $P = P(X_1, \dots, X_n)$ be a polynomial in $\mathbb{F}[X_1, \dots, X_p]$. Suppose that the coefficient of a monomial $X_1^{k_1} \dots X_p^{k_p}$, where each k_i is a non-negative integer, is non-zero in P and the degree $\deg(P)$ of P equals $\sum_{i=1}^p k_i$.*

If moreover S_1, \dots, S_p are any subsets of \mathbb{F} with $|S_i| > k_i$ for $i = 1, \dots, p$, then there are $s_1 \in S_1, \dots, s_p \in S_p$ so that $P(s_1, \dots, s_p) \neq 0$.

Let v_1, \dots, v_p be the vertices of $V(G) \setminus V(H)$ and v_{p+1}, \dots, v_n be the vertices of H . For each vertex $v_i \in V(G) \setminus V(H)$, let $S_i = \{1, \dots, k\} \setminus \{c(u) : u \in N(v_i) \cap V(H)\}$. By definition, only the colors of S_i can be assigned to v_i without creating a conflict with a neighbor of v_i in H . Let P be the following polynomial:

$$P(X_1, \dots, X_p) = \prod_{i=1}^p \prod_{\substack{v_j \in N(v_i) \\ j < i}} (X_i - X_j).$$

Suppose that c' is an extension of c to G such that $c'(v_i) \in S_i$. Note that c' is proper if and only if for every integer i and j such that $j < i \leq p$ and $v_i v_j \in E(G)$, $c'(i) - c'(j) \neq 0$. In particular c' is proper if and only if $P(c'(v_1), \dots, c'(v_p)) \neq 0$.

Let us suppose that we can apply the Combinatorial Nullstellensatz: suppose that P has a maximal non-zero monomial $X_1^{k_1} \dots X_p^{k_p}$ where $\sum_{i=1}^p k_i = \deg(P)$ and that each k_i verifies $k_i < |S_i|$. By the Combinatorial Nullstellensatz, there are $s_1 \in S_1, \dots, s_p \in S_p$ such that $P(s_1, \dots, s_p) \neq 0$.

Now let c' be an extension of c to G defined by $c'(v_i) = s_i$ for every $i \leq p$. Note that by definition of the s_i 's, $P(c'(v_1), \dots, c'(v_p)) \neq 0$ and thus c' is proper. Hence we were able to extend c to G .

Note that the previous procedure relies on two parts. The first consists in finding the sets S_i 's and the polynomial P . While finding the sets is often easy, it is the set of colors which do not create a conflict, it might be more difficult to find the polynomial depending on the problem. In this thesis, all polynomials are fairly simple.

The second step is finding the right monomial in P and showing that it has a non-zero coefficient. Remark that it is likely that no such monomial exists in P , in this case, one has to start again by changing the subgraph H . When the polynomial is of constant size (*i.e.* when we want to remove a fixed structure like a triangle), we can just exhibit the monomial and its coefficient. When the polynomial can have an arbitrary size, we must find manually a monomial and find a formula to show that it is non-zero.

We use the Combinatorial Nullstellensatz in Chapter 7 and Chapter 10.

Part I
Signed Graphs

Chapter 2

Introduction to signed graphs

The subject of this first part is the study of different problems on signed graphs. The goal of this chapter is to present what are signed graphs and to provide tools to manipulate them.

Signed graphs are a type of graph with two types of edges: positive edges and negative edges. They were introduced by Heider in [97] and latter formalized by Harary in [92] to model problems in social psychology [40, 178]. In these applications, the vertices are actors in a social environment (people in a given community, countries...) and the two types of edges model some notion of friendship (people who like/dislike each over, alliances and political tension for countries...). For more details about the diverse applications of signed graphs, we refer the reader to Section 2.1.4.

Contents

2.1	Key definitions	54
2.1.1	Definition of signed graphs	54
2.1.2	Balance of cycles, equivalence of signed graphs	55
2.1.3	Switching	55
2.1.4	Applications of signed graphs	58
2.2	Homomorphisms and coloring of signed graphs	58
2.2.1	Homomorphisms of signed graphs	59
2.2.2	Coloring of signed graphs	60
2.2.3	Sign-preserving homomorphisms and sign-preserving colorings	61
2.2.4	Signed cliques	62
2.3	Classes of cycles	63
2.4	Complexity	64
2.5	Similar notions and generalizations	65

We mainly follow the terminology of [153]. Most definitions in this chapter come from [92, 154, 208].

2.1 Key definitions

2.1.1 Definition of signed graphs

Let us start with the formal definition of a signed graph.

Definition 2.1 (Signed graph). A *signed graph* (G, σ) is a graph G , called the *underlying graph* of (G, σ) , along with a function $\sigma : E(G) \rightarrow \{+1, -1\}$ called the *signature* of (G, σ) . For every edge e of G , $\sigma(e)$ is the *sign* of e . For the sake of simplicity, we sometimes use $+$ (resp. $-$) instead of $+1$ (resp. -1) for the sign of an edge. The edges in $\sigma^{-1}(+1)$ are the *positive edges* of (G, σ) and the edges in $\sigma^{-1}(-1)$ are the *negative edges* of (G, σ) .

If the underlying graph G of a signed graph (G, σ) has some property (*e.g.* G is planar, simple, bipartite, complete...) then we say that (G, σ) has this property (*e.g.* (G, σ) is planar, simple, bipartite, complete...). A signed graph (G, σ) is *all-positive* (resp. *all-negative*) if it does not have negative edges (resp. positive edges). Unless stated otherwise, signed graphs are simple. See Figure 2.1 for examples of signed graphs. When drawing signed graphs, we always represent positive edges with blue edges and negative edges with discontinuous red edges.

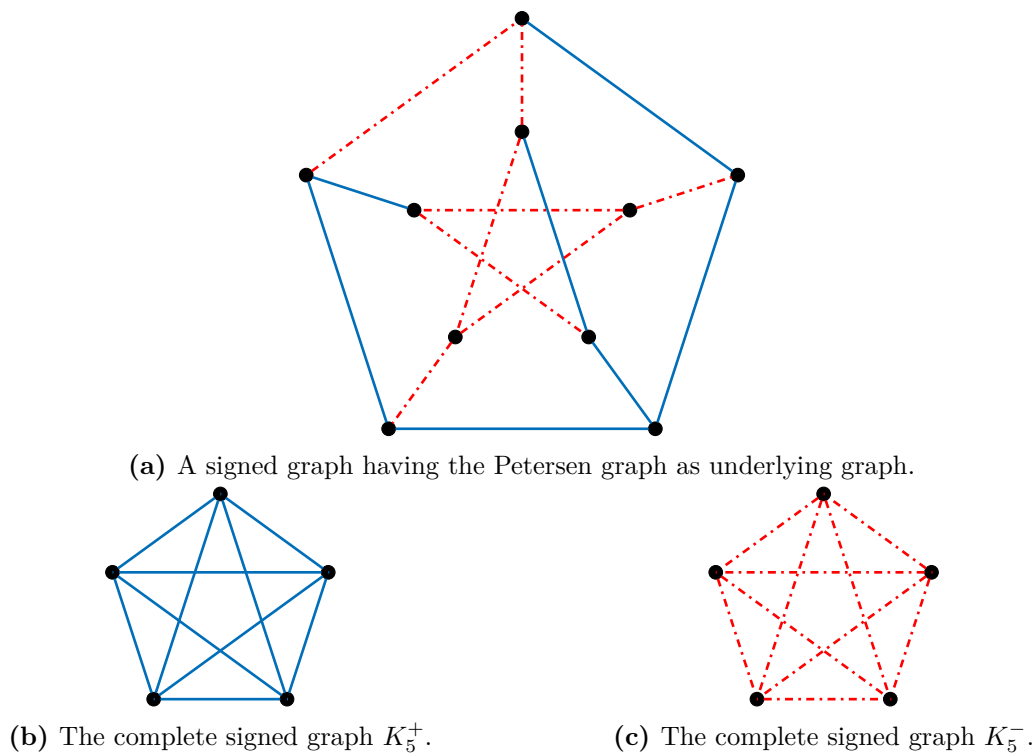


Figure 2.1: Examples of signed graphs.

Notation 2.2. We often denote a signed graph (G, σ) as (G, Σ) where Σ is the set of negative edges, that is $\Sigma = \sigma^{-1}(-1)$. This allows us to easily state the signature of the signed graph. These two ways to represent a signed graph are equivalent and will be used interchangeably.

We note K_p^+ (resp. K_p^-) for the complete signed graph (K_p, \emptyset) (resp. $(K_p, E(K_p))$) of order p with only positive (resp. negative) edges. See Figure 2.1(b) and Figure 2.1(c) for examples of complete signed graphs.

2.1.2 Balance of cycles, equivalence of signed graphs

One key concept of the theory of signed graphs is the notion of balance.

Definition 2.3 (Balance). Let (G, σ) be a signed graph and W be a walk s_0, \dots, s_n in G . We say that W is a *balanced walk* if $\sigma(W) = \sigma(s_0s_1)\sigma(s_1s_2)\dots\sigma(s_{i-1}s_i)\dots\sigma(s_{n-1}s_n) = 1$ and an *unbalanced walk* otherwise. Similarly, this notion can be extended to closed walks, paths and cycles.

A signed graph where all closed walks are balanced is said to be *balanced*. A signed graph (G, σ) , such that $(G, -\sigma)$ is balanced, is *antibalanced* ($-\sigma$ is the function which assign to each edge e of G the sign $-\sigma(e)$). In general, for the same ordinary graph G , there are several signatures σ for which (G, σ) is balanced.

We note an unbalanced path (resp. balanced path) of order k by UP_k (resp. BP_k) and an unbalanced cycle (resp. balanced cycle) of order k by UC_k (resp. BC_k). Note that there are multiple signed cycles/paths with the same length and the same balance. These notations refer to any cycle/path with those characteristics.

These notions of balanced and antibalanced graphs were introduced by Harary in [92]. See Figure 2.2 for an example of a balanced signed Petersen graph with some negative edges.

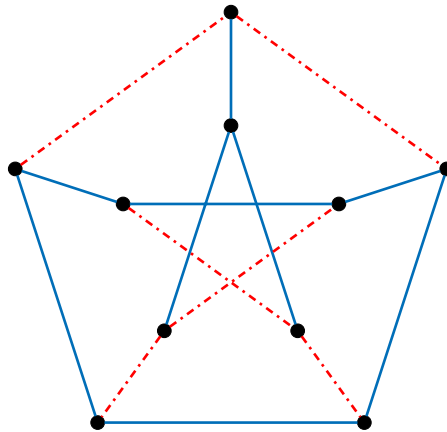


Figure 2.2: A balanced signed Petersen graph.

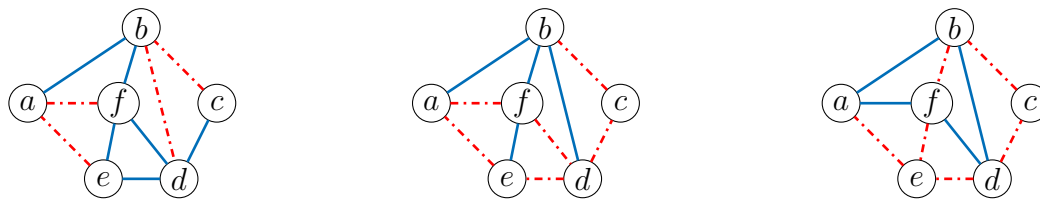
2.1.3 Switching

The following important operation on signed graphs was introduced by Zaslavsky in [208].

Definition 2.4. Let (G, σ) be a signed graph and v be a vertex of G . To *switch* v is to create the signed graph (G, σ') where $\sigma'(e) = -\sigma(e)$ when e is incident to v and $\sigma'(e) = \sigma(e)$ otherwise. To *switch* a set X of vertices of (G, σ) is to create the signed graph (G, σ') where σ' is obtained by switching every vertex of X , in any order (it is not difficult to see that the order does not matter).

See Figure 2.3 for some examples.

Lets us state some easy observations. Switching a vertex v of (G, σ) twice does nothing. Switching a set X of vertices of (G, σ) creates the signed graph (G, σ') where for every edge uv of G , $\sigma(uv) = -\sigma'(uv)$ if and only if one of u and v belongs to X and the other



(a) The signed graph (G, σ) . (b) The signed graph (G, σ') . (c) The signed graph (G, σ'') .

Figure 2.3: Example of switchings. The signed graph (G, σ') is obtained from (G, σ) by switching d . The signed graph (G, σ'') is obtained from (G, σ') by switching f . Alternatively, (G, σ'') is obtained from (G, σ) by switching $\{f, d\}$.

does not belong to X . Switching a set X of vertices of (G, σ) creates the same signed graph as switching the set $V(G) \setminus X$. The previous observations imply the following remark.

Remark. The switching operation on a signed graph (G, σ) consists exactly in choosing an edge cut $E(X, V(G) \setminus X)$ of G and negating the signs of all the edges of $E(X, V(G) \setminus X)$.

Zaslavsky in [208] defined the notion of equivalent signed graphs.

Definition 2.5 (Equivalence of signed graphs). Two signed graphs (G, σ_1) and (G, σ_2) on the same underlying graph G are *equivalent* if and only if we can obtain (G, σ_2) from (G, σ_1) by switching a subset of vertices of (G, σ_1) . In this case, we note $(G, \sigma_1) \equiv (G, \sigma_2)$. We also say that the two signatures σ_1 and σ_2 are equivalent and we note $\sigma_1 \equiv \sigma_2$.

One can observe that switching does not change the balance of closed walks. This follows from the following observation.

Observation 2.6 (Zaslavsky [208]). *If C is a cycle of a graph G , then switching any number of vertices of G does not change the sign of C .*

This implies that every signed graph obtained from (G, σ) by switching a subset of vertices has the same set of balanced (resp. unbalanced) closed walks. In fact, the set of signed graphs on the underlying graph G with the same set of balanced (resp. unbalanced) closed walks as (G, σ) is exactly the equivalence class of (G, σ) .

Theorem 2.7 (Zaslavsky [208]). *Two signed graphs on the same underlying graph G are equivalent if and only if they have the same set of balanced cycles.*

The previous theorem implies that we can work with the balance of closed walks or with switchings depending on which notion is the easiest to use when treating equivalence of signed graphs.

As mentioned before, two signed graphs (G, σ_1) and (G, σ_2) can both be balanced even if $\sigma_1 \neq \sigma_2$. Nonetheless, as highlighted by their equivalence, these two signed graphs have similar properties.

Let us see some examples. The signed Petersen graph in Figure 2.2 is equivalent to the signed Petersen graph with only positive edges. Two signed cycles of the same length are equivalent if they have the same parity of positive (resp. negative) edges. In particular all paths (resp. cycles) with the same length (resp. the same length and the same balance) are equivalent. Two signed forests with the same underlying graph are equivalent. See Figure 2.4 for one more example.

The following theorem follows from the proof of Theorem 2.7.

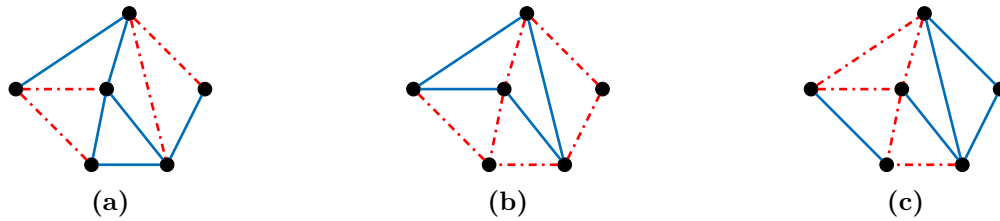


Figure 2.4: Three equivalent signed graphs.

Theorem 2.8 (Zaslavsky [208]). *For any two signed graphs (G, σ_1) and (G, σ_2) on the same underlying graph G , we can test in $O(|E(G)|)$ time whether they are equivalent or not.*

Independently, Harary and Kabell proposed an algorithm to determine if a graph is balanced using similar techniques [94].

In order to understand how Zaslavsky's algorithm works, we need to understand how to switch a signed tree so that the resulting signed tree is all-positive. Let (T, π) be our signed tree. Suppose that uv is a negative edge of (T, π) . Let V_u be the set of vertices connected to the vertex u in $T - uv$. Switching V_u modifies the sign of the edge cut $E(V_u, V(T) \setminus V_u)$. As T is a tree, only uv changes sign and becomes positive. Hence we were able to reduce the number of negative edges in our tree. By repeating this process, we can create the all-positive signed tree (T, \emptyset) .

Now consider that we are given two signed graphs (G, σ_1) and (G, σ_2) on the same underlying graph G . W.l.o.g. we assume that G is connected, otherwise we could perform the same procedure on each connected component. Choose a spanning tree T in G and perform switchings on both graphs in such a way that T becomes all-positive. We obtain (G, σ'_1) and (G, σ'_2) after the last step. If $\sigma'_1 = \sigma'_2$ then (G, σ'_1) and (G, σ'_2) are equivalent, otherwise they are not. Indeed, if an edge uv verifies $\sigma'_1(uv) \neq \sigma'_2(uv)$ then $uv \notin E(T)$. Consider the unique positive path in T joining u and v , the cycle composed of this path and the edge uv is balanced in one signed graph and unbalanced in the other, hence the two graphs are not equivalent.

Using a spanning tree is a powerful technique when studying signed graphs. We use this technique explicitly and implicitly in Chapter 4 and Chapter 5.

2.1.4 Applications of signed graphs

As mentioned in the introduction, signed graphs were designed to model social relationships. The sign of the edge between two vertices models the relationship between the vertices. A classical example of this is the study of the relationships between countries involved in both World wars [7].

A central notion of the theory of signed graphs, the balance of cycles, takes a particular meaning in social studies: this notion characterizes stable situations from unstable situations. An example, in the context of countries, can be described as follows: consider three countries A , B and C , B is allied with A and C while A and C are about to declare war. In this example, the country B will side with one of A or C , thus changing the nature of their relationship. The same situation but where A and B are already at war, is stable since the war declaration between A and C will not change the relation between A and B nor between B and C . Stable situations are represented by balanced cycles while unbalanced cycles represent unbalanced situations.

Another notion, called the *frustration index* of a signed graph, is the minimum number of edges to remove from a signed graph (G, σ) in order to obtain a balanced signed graph. This notion was introduced by Harary [93] under the name of *line index* of a signed graph. This notion has been extensively studied (*e.g.* see [67, 94, 111, 149]) and has applications in particular in physics. One such application is for ferromagnetic materials under the Ising model [182, 193]: each vertex has spin up or down, these vertices can switch spin (*i.e.* exchange up with down) which also changes the interaction between the vertices. As the state with the least negative edges, which can be obtained by switching, corresponds to a minimisation of the energy of the system, these materials tend to converge to this state. Unfortunately, computing the frustration index is NP-hard [111].

2.2 Homomorphisms and coloring of signed graphs

As we will see, there are multiple definitions of coloring for signed graphs. One of those colorings was introduced by Zaslavsky in [208]. In this thesis, we will not consider this coloring.

The one we are the most interested in is the notion of coloring presented by Naserasr, Rollová and Sopena in [154]. This notion relies on the generalization of the concept of homomorphism in the context of signed graphs.

2.2.1 Homomorphisms of signed graphs

Let us see how the concept of homomorphism is defined on signed graphs.

Definition 2.9 (Homomorphism of signed graphs). A *homomorphism* from a signed graph (G, σ) to a signed graph (H, π) is a homomorphism φ from G to H which maps balanced (resp. unbalanced) closed walks of (G, σ) to balanced (resp. unbalanced) closed walks of (H, π) .

Alternatively, a homomorphism φ from (G, σ) to (H, π) is a homomorphism from G to H such that there exists a signature σ' of G with $\sigma' \equiv \sigma$, such that if uv is an edge of G , then $\pi(\varphi(u)\varphi(v)) = \sigma'(uv)$.

When there is a homomorphism from (G, σ) to (H, π) , we note $(G, \sigma) \rightarrow_s (H, \pi)$ and say that (G, σ) *maps to* (H, π) . Here (H, π) is the *target graph* of the homomorphism.

See Figure 2.5 for an example of homomorphism of signed graphs.

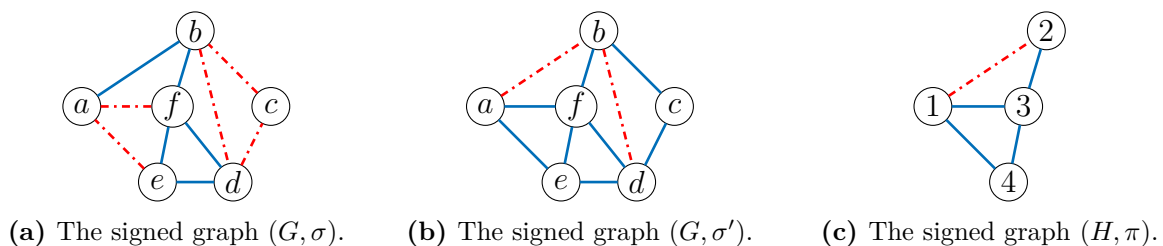


Figure 2.5: An example of a homomorphism φ from a signed graph (G, σ) to a signed graph (H, π) where $\varphi(a) = \varphi(d) = 1$, $\varphi(b) = 2$, $\varphi(f) = \varphi(c) = 3$ and $\varphi(e) = 4$. The signed graph (G, σ') is a signed graph equivalent to (G, σ) such that the edges of (G, σ') and their images through φ have the same sign.

Proposition 2.10 (Naserasr, Rollová and Sopena [154]). *The following statements hold.*

1. *If $(G, \sigma) \rightarrow_s (H, \pi)$, $(G, \sigma) \equiv (G, \sigma')$ and $(H, \pi) \equiv (H, \pi')$ then $(G, \sigma') \rightarrow_s (H, \pi')$,*
2. *if $(A, \sigma_A) \rightarrow_s (B, \sigma_B)$ and $(B, \sigma_B) \rightarrow_s (C, \sigma_C)$ then $(A, \sigma_A) \rightarrow_s (C, \sigma_C)$.*

Note that the previous proposition implies that, when trying to construct a homomorphism from (G, σ) to (H, π) , we can always switch the signed graph (H, π) in order to fix a simpler target graph.

Definition 2.11 (Chromatic number of signed graphs). Let (G, σ) be a signed graph. The *chromatic number* of (G, σ) , denoted $\chi_s(G, \sigma)$, is the order of the smallest simple signed graph (H, π) for which $(G, \sigma) \rightarrow_s (H, \pi)$.

Note that by definition, $\chi_s(G, \sigma) \geq \chi(G)$ for every signed graph (G, σ) .

Recall that we can construct homomorphisms of graphs through a sequence of identifications. In the rest of this thesis, a *digon* will be a UC_2 , *i.e.* two vertices linked by two edges, one positive and one negative. As we require the target graph in a homomorphism of signed graphs to be simple, we must be sure to never create digons nor loops when identifying vertices. A digon is created when we identify two vertices u and v which are endpoints of a UP_3 path.

Hence, before identifying two non-adjacent vertices u and v , we need to switch the signed graph in order to remove every UP_3 containing u and v . Note that this is not always possible. For example, in the unbalanced cycle UC_4 , we cannot identify any pair of vertices.

We say that two vertices u and v of a signed graph (G, σ) are *identifiable* if and only if there exist σ' equivalent to σ such that, in (G, σ') , u and v can be identified without creating a loop nor a digon. The following theorem gives a characterization of identifiable vertices.

Theorem 2.12 (Naserasr, Rollová and Sopena [154]). *Two vertices of a signed graph are identifiable if and only if they are not adjacent and do not belong to the same UC_4 .*

In Figure 2.5, we identified the two pairs of vertices $\{a, d\}$ and $\{c, f\}$ in (G, σ') to create (H, π) . Note that (H, π) does not contain any pair of identifiable vertices by Theorem 2.12.

2.2.2 Coloring of signed graphs

As for undirected graphs, we can define a notion of k -coloring for signed graphs. This leads to an alternative definition of the chromatic number of signed graphs.

Definition 2.13 (k -coloring of signed graphs). A signed graph (G, σ) admits a k -coloring if there exists $\sigma' \equiv \sigma$ such that (G, σ') admits a proper vertex coloring $c : V(G) \rightarrow \llbracket k \rrbracket$ verifying that for every $i, j \in \llbracket k \rrbracket$, all edges uv with $c(u) = i$ and $c(v) = j$ have the same sign in (G, σ') .

The *chromatic number* $\chi_s(G, \sigma)$ of (G, σ) is the smallest k such that (G, σ) admits a k -coloring.

As for ordinary graphs, one can create a homomorphism from a coloring: the colors are the vertices of the target graph. To construct a coloring of a signed graph (G, σ) from a homomorphism φ from (G, σ) to a signed graph (H, π) , it suffices to color the vertices of G with their image through φ and switch (G, σ) in such a way that each edge has the same sign as its image through the homomorphism (see Figure 2.6). In order to find which vertices to switch to achieve this signature, one can satisfy a particular 2-SAT formula with variables $(x_u)_{u \in V(G)}$ and switch the vertices assigned to true. Such a formula can be constructed as follows: for each edge $uv \in E(G)$, if uv has the same sign as its image through φ then add the two clauses $(x_u \vee \bar{x}_v)(\bar{x}_u \vee x_v)$, *i.e.* u and v must both be switched or both not be switched. If uv does not have the same sign as its image then add to the 2-SAT formula the two following clauses: $(\bar{x}_u \vee \bar{x}_v)(x_u \vee x_v)$, *i.e.* exactly one of u and v must be switched. Note that such a formula is always satisfiable by definition of a homomorphism.

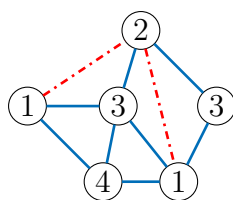


Figure 2.6: A coloring of a signed graph. This coloring is a coloring of the graph (G, σ') of Figure 2.5 obtained from the homomorphism presented in the figure.

Note that, if (G, σ) is a balanced (resp. antibalanced) signed graph then $\chi_s(G, \sigma) = \chi(G)$. Indeed if $G \rightarrow H$, then $(G, \sigma) \rightarrow_s (H, \emptyset)$ (recall that (H, \emptyset) is the all-positive signed graph with underlying graph H) as every closed walk of (H, \emptyset) is balanced.

Note that a k -coloring of (G, σ) is also a k -coloring of $(G, -\sigma)$. We can state a more precise statement with the following observation.

Observation 2.14. *Let (G, σ) and (H, π) be two signed graphs. We have $(G, \sigma) \rightarrow_s (H, \pi)$ if and only if $(G, -\sigma) \rightarrow_s (H, -\pi)$. In particular, $\chi_s(G, \sigma) = \chi_s(G, -\sigma)$.*

This observation is useful when trying to simplify the case analysis. In particular in Chapter 3, we reduce some complexity problems to others using this observation.

2.2.3 Sign-preserving homomorphisms and sign-preserving colorings

Let us see another important type of homomorphism and coloring of signed graphs.

Definition 2.15 (Sign-preserving homomorphism of signed graphs). A *sign-preserving homomorphism* from a signed graph (G, σ) to a signed graph (H, π) is a homomorphism φ from G to H such that for every edge $e \in E(G)$, $\pi(\varphi(e)) = \sigma(e)$. When there is a sign-preserving homomorphism from (G, σ) to (H, π) , we note $(G, \sigma) \rightarrow_s^p (H, \pi)$.

A signed graph (G, σ) admits a *sign-preserving k -coloring* if (G, σ) admits a proper vertex coloring $c : V(G) \rightarrow \llbracket k \rrbracket$ verifying that for every $i, j \in \llbracket k \rrbracket$, all edges uv with $c(u) = i$ and $c(v) = j$ have the same sign.

The *sign-preserving chromatic number* of a signed graph (G, σ) , denoted $\chi_s^p(G, \sigma)$, is the smallest order of a simple signed graph (H, π) such that $(G, \sigma) \rightarrow_s^p (H, \pi)$. Alternatively, it is the smallest k for which (G, σ) admits a sign-preserving k -coloring.

Note that a signed graph can be interpreted as a 2-edge-colored graph where the two colors are $+1$ and -1 . With this interpretation, a sign-preserving homomorphism is the same object as a homomorphism of 2-edge-colored graphs. This is why, in the literature, we often find the term 2-edge-colored graph to designate a signed graph for which we do not allow switchings. Note however that by specifying that the two colors are $+1$ and -1 in signed graphs, we obtain an object with more structure than just the two colors as we can reason on signs.

For example, with the notation of Figure 2.5, φ is a sign-preserving homomorphism from (G, σ') to (H, π) . Note that there is no sign-preserving homomorphism from (G, σ) to (H, π) .

The notion of sign-preserving homomorphism is related to homomorphisms of signed graphs through the following construction.

Definition 2.16 (Double switching graph). The *double switching graph* of (G, σ) , denoted $\text{DSG}(G, \sigma)$, is the signed graph constructed as follows.

1. The vertex set of $\text{DSG}(G, \sigma)$ is $V(G) \times \{0, 1\}$,
2. for every edge uv of G , we create the following four edges $(u, 0)(v, 0)$, $(u, 1)(v, 1)$, $(u, 0)(v, 1)$ and $(u, 1)(v, 0)$, the first two edges having sign $\sigma(uv)$ and the latter two edges having sign $-\sigma(uv)$.

See Figure 2.7 for an example of a double switching graph.

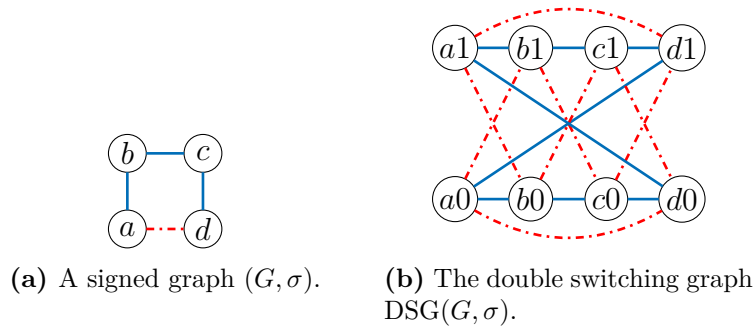


Figure 2.7: An example of a double switching graph. For simplicity, a vertex (x, ε) of $\text{DSG}(G, \sigma)$ with $x \in \{a, b, c, d\}$ and $\varepsilon \in \{0, 1\}$ is noted $x\varepsilon$.

The following theorem justifies such a construction.

Theorem 2.17 (Brewster, Foucaud, Hell and Naserasr [28]). *Let (G, σ) and (H, π) be two signed graphs. The following are equivalent.*

1. $(G, \sigma) \xrightarrow{s} (H, \pi)$,
2. $(G, \sigma) \xrightarrow{s} \text{DSG}(H, \pi)$ and,
3. $\text{DSG}(G, \sigma) \xrightarrow{s} \text{DSG}(H, \pi)$.

We can also construct sign-preserving homomorphism by a sequence of identifications. Here two vertices u and v can be identified if and only if they are not adjacent and if there is no vertex w in (G, σ) such that $\sigma(uw) \neq \sigma(wv)$. The path u, w, v is an alternating path. In general, a walk (resp closed walk) s_0, s_1, \dots, s_n is *alternating* if and only if $\sigma(s_i s_{i+1}) \neq \sigma(s_{i+1} s_{i+2})$, for every $i \in \llbracket 0, n-2 \rrbracket$ (resp. $i \in \llbracket 0, n-1 \rrbracket$).

2.2.4 Signed cliques

The notion of homomorphism allows us to generalize the notion of cliques.

Definition 2.18 (Signed clique). A signed graph (G, σ) is a *signed clique* if $\chi_s(G, \sigma) = |V(G)|$. The *absolute signed clique number* of a signed graph (G, σ) , denoted $\omega_{as}(G, \sigma)$ is the order of the largest subgraph of (G, σ) which is a signed clique. The *relative signed clique number* of a signed graph (G, σ) , denoted $\omega_{rs}(G, \sigma)$ is the greatest number of vertices which are pairwise non-identifiable in (G, σ) .

Note that $\chi_s(G, \sigma) \geq \omega_{rs}(G, \sigma) \geq \omega_{as}(G, \sigma)$. Signed cliques are interesting as it is easy to verify whether a given signed graph is a signed clique or not (see Theorem 2.12) and their chromatic number is very easy to compute. Because of this, they make useful examples for bounds on the chromatic number. See Figure 2.8(a) and Figure 2.8(b) for some examples.

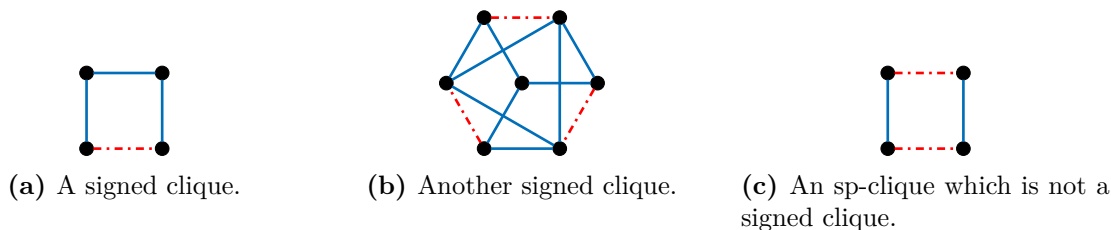


Figure 2.8: Two examples of a signed cliques (a) and (b) and an example of an sp-clique (c).

Other interesting cliques in signed graphs are sp-cliques.

Definition 2.19 (sp-clique). A signed graph (G, σ) is an *sp-clique* if $\chi_s^p(G, \sigma) = |V(G)|$.

An sp-clique is the same concept as a 2-edge-colored clique. See Figure 2.8(c) for an example.

One can easily transform an sp-clique into a signed clique by creating a new vertex adjacent to all the other vertices of the sp-clique with positive edges.

2.3 Classes of cycles

One important part of the study of signed graphs is to study the signed cycles. As we saw, a signed cycle can be balanced or unbalanced. Another, more usual, way to divide cycles is through the parity of their lengths. Indeed even cycles behave differently than odd cycles.

This implies that we can separate the set of all signed cycles into four families BC_{even} , BC_{odd} , UC_{even} and UC_{odd} , depending on the parity of the number of negative edges (even for BC_{even} and BC_{odd} and odd for UC_{even} and UC_{odd}) and the parity of the length of the cycle (even for BC_{even} and UC_{even} and odd for BC_{odd} and UC_{odd}). See Figure 2.9(a) for some examples.

Note that every graph has balanced even closed walks as long as it is non-empty. Indeed, if uv is an edge of a signed graph (G, σ) , the closed walk uvu is balanced even. The class of graphs with only signed cycles in BC_{even} is the class of bipartite balanced signed graphs, *i.e.* signed graphs which are bipartite (they do not have odd cycles) and

	even length	odd length
balanced	BC_{even} 	BC_{odd}
unbalanced	UC_{even} 	UC_{odd}

(a) The usual division between balanced/unbalanced and even/odd for signed cycles with some examples.

number of edges	even number of positive	odd number of positive
even number of negative	BC_{even}	BC_{odd}
odd number of negative	UC_{odd}	UC_{even}

(b) The same classes divided according to the parity of the number of edges of each sign.

Figure 2.9: Classes of signed cycles.

balanced (all of their cycles are balanced). Note that a bipartite balanced signed graph (G, σ) verifies $\chi_s(G, \sigma) = \chi(G) = 2$ since every such graph is balanced, and thus equivalent to (G, \emptyset) which verifies $\chi_s(G, \emptyset) = \chi(G)$.

The most difficult class to manipulate is the class UC_{even} . Indeed this class contains the signed graph UC_4 which forbids identifications of some non-adjacent vertices (see Theorem 2.12).

This intuition is confirmed when looking at the chromatic number of signed cycles.

Theorem 2.20. *Let (C, σ) be a signed cycle. We then have:*

1. $\chi_s(C, \sigma) = 2$ if $(C, \sigma) \in BC_{even}$,
2. $\chi_s(C, \sigma) = 3$ if $(C, \sigma) \in BC_{odd} \cup UC_{odd}$,
3. $\chi_s(C, \sigma) = 4$ if $(C, \sigma) \in UC_{even}$.

Proof. By [60], we already have the upper bounds. A homomorphism of signed graphs is also a homomorphism of graphs thus $\chi(C) \leq \chi_s(C, \sigma)$. This proves the lower bounds for the first two cases. Let $(C, \sigma) \equiv UC_{2q}$ and suppose $\chi_s(C, \sigma) \leq 3$. Then $(C, \sigma) \rightarrow_s (K_3, \pi)$. In each case, (K_3, π) can be switched either to be all-positive or to be all-negative. This means that (C, σ) can be switched either to be all-positive or to be all-negative, which is not the case as UC_{2q} has an odd number of negative edges and an odd number of positive edges, a contradiction. We get the desired lower bounds in each case. \square

The table in Figure 2.9(b) gives a more accurate representation of the division between these classes. The “more complicated” class UC_{even} is the furthest away from the simplest, BC_{even} , and we can see the symmetry between BC_{odd} and UC_{odd} . This representation is important to keep in mind, as at first glance, UC_{odd} seems to be the more complex class while in reality, UC_{even} is.

We can define the equivalent of a subgraph for signed graphs.

Definition 2.21 (Signed subgraph). A *signed subgraph* $(H, \sigma[H])$ of a signed graph (G, σ) is a signed graph such that H is a subgraph of G and for every edge e of H , $\sigma[H](e) = \sigma(e)$. For a subset X of vertices of G , the *induced signed subgraph* $(G, \sigma)[X]$ of (G, σ) (or signed subgraph of (G, σ) induced by X), is the signed subgraph $(G[X], \sigma[X])$ of (G, σ) for which $\sigma[X] = \sigma[G[X]]$.

2.4 Complexity

Let us define two decision problems. Let (H, π) be a fixed signed graph.

SIGNED- (H, π) -COLORING

Input: A signed graph (G, σ) .

Question: Do we have $(G, \sigma) \rightarrow_s (H, \pi)$?

(H, π) -COLORING

Input: A signed graph (G, σ) .

Question: Do we have $(G, \sigma) \rightarrow_s^p (H, \pi)$?

The *switching core* (s-core for short) of a signed graph (G, σ) is the smallest subgraph (H, π) of (G, σ) for which $(G, \sigma) \rightarrow_s (H, \pi)$. A signed graph (G, σ) is a *switching core* if (G, σ) is its own switching core. The *switch-preserving core* (sp-core for short) of a signed graph (G, σ) is the smallest subgraph (H, π) of (G, σ) for which $(G, \sigma) \rightarrow_s^p (H, \pi)$. A signed graph (G, σ) is an *switch-preserving core* if (G, σ) is its own sp-core.

We have the following characterization for the complexity of SIGNED- (H, π) -COLORING.

Theorem 2.22 (Brewster, Foucaud, Hell and Naserasr [28] and Brewster and Siggers [29]). *Let (H, π) be a signed graph. SIGNED- (H, π) -COLORING is in P if the s-core of (H, π) has at most two edges, and is NP-complete otherwise.*

To this date, there is no similar characterization for (H, π) -COLORING.

2.5 Similar notions and generalizations

There are many parallels between the study of signed graphs and the study of oriented graphs. In both cases, we have two types of edges: positive and negative for signed graphs and two orientations for oriented graphs. The notion of homomorphism of oriented graphs has many parallels with the notion of sign-preserving homomorphisms.

Oriented graphs also have their own switching operation.

Definition 2.23 (Pushable oriented graphs). Let \vec{G} be an oriented graph. *Pushing* a vertex v of \vec{G} consists in inverting the orientation of all the edges incident with v . Two oriented graphs are equivalent if one can transform one into the other by a sequence of pushings. A *pushable oriented graph* \vec{G} is an equivalence class of this equivalence relation.

From these notions, we can derive the corresponding homomorphism and coloring notions. It is often interesting to look at oriented graphs when dealing with signed graphs as they have similar behaviors. See [175] for more details on oriented coloring.

A way to generalize signed graphs is to allow the signs to belong to any group.

Definition 2.24 (Gain graph). A *gain graph* is a graph where the edges are given an orientation and labelled by elements of a group \mathcal{G} . If the edge uv is labelled by g then vu is labelled by g^{-1} , the inverse of g .

We could define a notion of switching on gain graphs, this would correspond to multiplying all the edges incident with a vertex v by some element of \mathcal{G} (by taking the

orientation into account). Note that a signed graph is a gain graph where the group is $(\{-1, 1\}, \times)$. Note that for gain graphs most of the useful properties of signed graphs are not guaranteed. For example, with such a definition of switching, if G is not abelian then the switching operation is not commutative. Moreover, some cycles with different products may be equivalent, for example it is the case for every triangle labelled with the group $(\{0, 1, 2\}, +)$.

One particular case which preserves many properties of signed graphs is the case of gain graphs with groups of the form $(\{0, 1\}^p, \oplus)$ where p is a positive integer and \oplus is the bit-wise addition. They correspond to graphs where each edge is assigned multiple signs.

Another generalization, called *biased graphs*, has properties similar to signed graphs (for more details see [209, 210]).

Chapter 3

Complexity of edge-colored and signed graphs modification problems

Graph coloring problems such as k -COLORING are among the most fundamental problems in algorithmic graph theory. The problem H -COLORING is a homomorphism-based generalization of k -COLORING that is extensively studied [35, 71, 98, 150].

In this chapter, we consider parameterized variants of H -COLORING (resp. SIGNED- (H, π) -COLORING) where H is an edge-colored graph (resp. (H, π) is a signed graph). We allow loops and multiple edges, but multiple edges of the same color are irrelevant in H (resp. (H, π)).

For edge-colored graphs H , the H -COLORING problems are well-studied, see for example [16, 26, 25, 27, 28]. They are special cases of *Constraint Satisfaction Problems* (CSPs). A large set of CSPs can be modeled by homomorphisms from general relational structures to a fixed relational structure H [71]. The corresponding decision problem is noted as H -CSP. When H has only binary relations, H can be seen as an edge-colored graph (a relation corresponds to the set of edges of a given color) and H -CSP is exactly H -COLORING. The complexity of H -CSP has been the subject of intensive research in the last decades, since Feder and Vardi conjectured in [71] that H -CSP is either in P or is NP -complete — a statement that became known as the Dichotomy Conjecture. The latter conjecture was recently solved in [33, 214] independently; the criterion for H -CSP to be in P is based on certain algebraic properties of H . Nevertheless, determining whether a structure H satisfies this criterion is not an easy task (even for targets as simple as oriented trees [35]). Thus, the study of more simple and elegant complexity classifications for relevant special cases is of high importance.

The complexity of H -COLORING when H is uncolored is well-understood: it is in P if H contains a loop or is bipartite; otherwise it is NP -complete [98]. This was one of the early dichotomy results in the area. On the other hand, when H is a 2-edge-colored graph, it was proved that the class of H -COLORING problems captures the difficulty of the whole class of H -CSP problems [28], and thus the dichotomy classification for this class of problems is expected to be much more intricate.

Our goal is to study generalizations of H -COLORING problems for edge-colored graphs by enhancing them as *modification problems*. In this setting, given a graph property \mathcal{P} and a graph operation π , the graph modification problem for \mathcal{P} and π asks whether an input graph G can be made to satisfy property \mathcal{P} after applying operation π a given number k of times. This is a classic setting studied extensively both in the realms of classical and parameterized complexity, see for example [37, 52, 129, 141, 205]. In this

context, the most studied graph operations are vertex-deletion and edge-deletion, see the seminal papers [141, 205].

For a fixed graph H , let $\mathcal{P}(H)$ denote the property of admitting a homomorphism to H . Certain standard computational problems can be stated as graph modification problems for $\mathcal{P}(H)$. For example, VERTEX COVER is the graph modification problem for property $\mathcal{P}(K_1)$ and operation vertex-deletion. Similarly, ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION are the graph modification problems for $\mathcal{P}(K_2)$ and vertex-deletion, and $\mathcal{P}(K_2)$ and edge-deletion, respectively.

When considering signed graphs (which can be viewed as edge-colored graphs with only two edge-colors), another operation of interest is switching. Switching a vertex of a signed graph transforms a signed graph into another, therefore we can view switching as a modification operation on signed graphs for the (H, π) -COLORING problem.

Signed graph can be manipulated with two types of homomorphisms: sign-preserving homomorphisms, in which case they behave like 2-edge-colored graphs, or homomorphisms of signed graphs, for which switching is unlimited. This lead us to also consider modification problems (vertex deletion and edge deletion) for SIGNED- (H, π) -COLORING.

Let us now formally define the problems we will consider (the parameter is always k).

<p>VERTEX DELETION H-COLORING</p> <p>Input: An edge-colored graph G, an integer k.</p> <p>Question: Is there a set S of at most k vertices of G such that $(G - S) \rightarrow H$?</p>	<p>Parameter: k.</p>
---	--

<p>EDGE DELETION H-COLORING</p> <p>Input: An edge-colored graph G, an integer k.</p> <p>Question: Is there a set S of at most k edges of G such that $(G - S) \rightarrow H$?</p>	<p>Parameter: k.</p>
--	--

<p>LIMITED SWITCHINGS (H, π)-COLORING</p> <p>Input: A signed graph (G, σ), an integer k.</p> <p>Question: Is there a set S of at most k vertices of G such that the signed graph (G, σ') obtained from (G, σ) by switching every vertex of S satisfies $(G, \sigma') \rightarrow_s^p (H, \pi)$?</p>	<p>Parameter: k.</p>
--	--

<p>VERTEX DELETION SIGNED-(H, π)-COLORING</p> <p>Input: A signed graph (G, σ), an integer k.</p> <p>Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S \rightarrow_s (H, \pi)$?</p>	<p>Parameter: k.</p>
---	--

<p>EDGE DELETION SIGNED-(H, π)-COLORING</p> <p>Input: A signed graph (G, σ), an integer k.</p> <p>Question: Is there a set S of at most k edges of G such that $(G, \sigma) - S \rightarrow_s (H, \pi)$?</p>	<p>Parameter: k.</p>
--	--

In the study of the five above problems, one may assume that H (resp. (H, π)) is some kind of core (edge-colored core, resp. sp -core or s -core) depending on the nature of the base problem. Indeed, our target graph H (resp. (H, π)) being a constant of the problem, we can transform it into its core in constant time. Note that if the target was part of the input then computing its core (for whatever notion of core that we chose) would be NP-complete [99].

Of course, if one of the five problems is NP-complete even for $k = 0$, then the problem for general k is NP-complete and not in XP (unless $P = NP$). This is for example the

case when H is an edge-colored graph containing only a monochromatic triangle: then we have 3-COLORING for $k = 0$ in the first three problems. Thus, from the point of view of parameterized complexity, it is of primary interest to consider these problems for edge-colored graphs H (resp. signed graphs (H, π)) such that the problem for $k = 0$ is in P . In that case a simple brute-force algorithm iterating over all k -subsets of vertices of G implies that the five problems are in XP and hence the interesting question is whether these problems are FPT or not. For undirected graphs, the only cores H for which H -COLORING is in P are the three connected graphs with at most one edge [98] (a single vertex with no edge, a single vertex with a loop, two vertices joined by an edge), so in that case the interest of these problems is limited. However, for many interesting families of edge-colored graphs H , the problem H -COLORING is in P , and the class of such graphs H is not very well understood, see [26, 25, 27]. Even when H is a 2-edge-colored cycle, a 2-edge-colored tree or a 2-edge-colored complete graph, there are infinitely many H 's for which H -COLORING is NP -complete and infinitely many H for which H -COLORING is in P [26]. For signed graphs, the complexity of SIGNED- (H, π) -COLORING is completely characterized (see Theorem 2.22).

Related work. Several works address the parameterized complexity of graph coloring problems. Graph coloring problems parameterized by structural parameters are considered in [118]. In [48], the vertex-deletion variant of H -LIST-COLORING is studied. Graph modification problems for COLORING in specific graph classes and for operations vertex-deletion and edge-deletion are considered, for example in [38] (bipartite graphs, split graphs) and [180] (comparability graphs).

Every problem VERTEX DELETION H -COLORING can be encoded as a special *weighted homomorphism* problem H' -WEIGHTED-COLORING, as considered in [158]. In that setting, the target H' is a graph with integer weights, and the goal is to find a homomorphism from some input graph G whose weight (*i.e.* the sum of weights of the images of the vertices of G) is at most some given integer k' . In our setting, we could generalize this problem to edge-colored graphs and build H' from H by setting all weights to 0 and adding a new vertex x adjacent to all vertices of H with weight 1. Now, finding a weighted homomorphism from G to H with weight at most k is the same as having a positive solution to VERTEX DELETION H -COLORING (vertices mapped to x represent the deleted vertices in S). A similar notion was studied for general CSPs in [34]. In that setting, only one “free” target vertex has weight 0 and all the others, weight 1, and the goal is to find a homomorphism of weight at most a given integer k . The Boolean CSP version where there are only two target values, 0 and 1, and we wish to minimize the number of variables set to 1, is called the MIN ONES problem [128].

Algorithmic problems relative to the operation of *Seidel switching*, similar to our switching, have been considered. Given an undirected graph G , the Seidel switching operation performed at a vertex exchanges all adjacencies and non-adjacencies of v . This can be seen as performing a switching operation in a 2-edge-colored complete graph, where blue edges are the actual edges of G , and red edges are its non-edges. In [63, 119], the complexity of graph modification problems with respect to the Seidel switching operation and the property of being a member of certain graph classes has been studied. Our work on LIMITED SWITCHINGS H -COLORING problems can be seen as a variation of these problems, generalized to arbitrary 2-edge-colored graphs.

Our results. We study the classical and parameterized complexities of the five problems VERTEX DELETION H -COLORING, EDGE DELETION H -COLORING, LIMITED SWITCHINGS (H, π) -COLORING, VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING. For the first three problems, our focus is on t -edge-colored graphs H of order at most 2 where t is an integer ($t = 2$ and we see H as a signed graph for LIMITED SWITCHINGS (H, π) -COLORING). Despite having only two vertices, H -COLORING for such an H is interesting and non-trivial; it is proved to be in \mathbf{P} by two different non-trivial methods, see [16, 27]. Thus, the three considered problems are in \mathbf{XP} for such an H . (Note that for suitable 1-edge-colored graphs H of order 1 or 2, VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING include VERTEX COVER and ODD CYCLE TRANSVERSAL.)

We completely classify the classical complexity of VERTEX DELETION H -COLORING when H is a t -edge-colored graph of arbitrary order: it is either trivially in \mathbf{P} or \mathbf{NP} -complete. It turns out that all VERTEX DELETION H -COLORING problems are FPT when H has order at most 2. To prove this, we extend a method from [27] and reduce the problem to an FPT variant of 2-SAT.

For EDGE DELETION H -COLORING, a classical complexity dichotomy seems more difficult to obtain, as there are non-trivial polynomial cases. We perform such a classification when H is a t -edge-colored graph of order at most 2. Similar 2-SAT-based arguments as for VERTEX DELETION H -COLORING give a FPT algorithm for EDGE DELETION H -COLORING when H has order at most 2.

For LIMITED SWITCHINGS (H, π) -COLORING when (H, π) is a signed graph, the classical dichotomy is again more difficult to obtain. We perform such a classification by using some characteristics of the switch operation and by giving some reductions to well-known \mathbf{NP} -complete problems. In contrast to the two previous cases for the parameterized complexity, we show that for three signed graphs (H, π) of order 2, LIMITED SWITCHINGS (H, π) -COLORING is already $\mathbf{W}[1]$ -hard (and cannot be solved in time $f(k)|V(G)|^{o(k)}$ for any computable function f , assuming the ETH). For all other signed graphs of order 2, we prove that LIMITED SWITCHINGS H -COLORING is FPT.

For VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING, we completely classify the classical complexity of these problems, indeed the number of open cases is more limited than in the previous problems. Similar arguments as for VERTEX DELETION H -COLORING can be used for VERTEX DELETION SIGNED- (H, π) -COLORING while we treat the problem case by case for EDGE DELETION SIGNED- (H, π) -COLORING. For the parameterized complexity of these problems, we prove that for all signed graphs whose s -core contains at most two edges (*i.e.* signed graphs for which the problem with $k = 0$ is in \mathbf{P}), the problems VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING are FPT.

Table 3.1 presents a brief overview of our results, and Table 3.2 lists the classical and parameterized complexities of the five considered problems for all 2-edge-colored graphs (resp. signed graphs) of order at most 2.

Sections 3.1 through 3.3 are joint work with Florent Foucaud, Hervé Hocquard, Valia Mitsou and Théo Pierron. An extended abstract of Sections 3.1 to 3.3 was published [76] in the proceedings of the international conference IPEC 2019. A full version of the corresponding paper can be found on arXiv [77].

This work is partially supported by the ANR project HOSIGRA (ANR-17-CE40-0022) and the IFCAM project “Applications of graph homomorphisms” (MA/IFCAM/18/39).

Problem	P vs NP-hard	FPT vs. W[1]-hard when $ V(H) \leq 2$
VERTEX DELETION H -COLORING	Dichotomy for all graphs (Cor. 3.12)	All FPT (Thm. 3.20)
EDGE DELETION H -COLORING	Dichotomy when $ V(H) \leq 2$ (Thm. 3.13)	All FPT (Thm. 3.20)
LIMITED SWITCHINGS (H, π) -COLORING	Dichotomy when $ V(H) \leq 2$ (Thm. 3.22)	Dichotomy (Thms. 3.23, 3.24)
VERTEX DELETION SIGNED- (H, π) -COLORING	Dichotomy for all graphs (Thm. 3.26)	FPT if in XP (Thm. 3.26)
EDGE DELETION SIGNED- (H, π) -COLORING	Dichotomy for all graphs (Thm. 3.38)	FPT if in XP (Thm. 3.38)

Table 3.1: Overview of our main results, sorted by problem and by type of classification.













$H / (H, \pi)$	VERTEX DELETION H -COLORING	EDGE DELETION H -COLORING	LIMITED SWITCHINGS (H, π) -COLORING	VERTEX DELETION SIGNED- (H, π) -COLORING	EDGE DELETION SIGNED- (H, π) -COLORING
 $H_{r,b}^1$	P	P	P	P	P
 H_b^1	NP-h but FPT	P	P	NP-h but FPT	NP-h but FPT
 H_-^1	NP-h but FPT	P	P	NP-h but FPT	P
 $H_{r,b}^{2-}$	NP-h but FPT	P	P	NP-h but FPT	NP-h but FPT
 $H_{-,-}^{2b}$	NP-h but FPT	NP-h but FPT	P	NP-h but FPT	NP-h but FPT
 $H_{r,b}^{2b}$	NP-h but FPT	NP-h but FPT	NP-h but FPT	NP-h even for $k = 0$	NP-h even for $k = 0$
 $H_{r,-}^{2b}$	NP-h but FPT	NP-h but FPT	NP-h but FPT	Not an s-core	Not an s-core
 $H_{r,r}^{2b}$	NP-h but FPT	NP-h but FPT	P	Not an s-core	Not an s-core
 $H_{-,-}^{2rb}$	NP-h but FPT	NP-h but FPT	P	NP-h but FPT	NP-h but FPT
 $H_{r,b}^{2rb}$	NP-h but FPT	NP-h but FPT	NP-h and W[1]-h	NP-h even for $k = 0$	NP-h even for $k = 0$
 $H_{r,-}^{2rb}$	NP-h but FPT	NP-h but FPT	NP-h and W[1]-h	NP-h even for $k = 0$	NP-h even for $k = 0$
 $H_{r,r}^{2rb}$	NP-h but FPT	NP-h but FPT	NP-h and W[1]-h	NP-h even for $k = 0$	NP-h even for $k = 0$

Table 3.2: Our results for target 2-edge-colored graphs H (resp. signed graphs (H, π)) of order at most 2 (up to inversion of edge-colors (resp. signs), there are twelve such 2-edge-colored graphs (resp. signed graphs) to consider).

This chapter is structured as follows. In Section 3.1, we state some definitions and make some preliminary observations in relation with the literature. We also reformulate some particular instances of our problems to highlight well known complexity problems. Section 3.2 presents our study of VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING. The problem LIMITED SWITCHINGS (H, π) -COLORING is treated in Section 3.3. In Section 3.4, we analyse the problem VERTEX DELETION SIGNED- (H, π) -COLORING while Section 3.5 is for the problem EDGE DELETION SIGNED- (H, π) -COLORING. Finally, we conclude in Section 3.6.

Contents

3.1 Preliminaries	72
3.1.1 Some known complexity dichotomies	72
3.1.2 Homomorphism dualities and FPT time	74
3.1.3 Reformulating some modification problems	76
3.2 Edge-colored modification problems	77
3.2.1 VERTEX DELETION H -COLORING: P/NP-complete dichotomy	77
3.2.2 EDGE DELETION H -COLORING: P/NP-complete dichotomy when H has order 2	80
3.2.3 VERTEX/EDGE DELETION H -COLORING: FPT algorithms when H has order 2	84
3.3 Limited Switchings (H, π)-Coloring when H has order 2	87
3.3.1 LIMITED SWITCHINGS (H, π) -COLORING: P/NP-complete dichotomy	87
3.3.2 LIMITED SWITCHINGS (H, π) -COLORING: FPT cases	88
3.3.3 LIMITED SWITCHINGS (H, π) -COLORING: W[1]-hard cases	89
3.3.3.1 Generic reduction	90
3.3.3.2 Gadgets for $H_{r,r}^{2rb}$	91
3.3.3.3 Gadgets for $H_{r,-}^{2rb}$	92
3.3.3.4 Gadgets for $H_{r,b}^{2rb}$	93
3.4 Vertex Deletion Signed-(H, π)-Coloring	94
3.4.1 VERTEX DELETION SIGNED- $H_{,-}^{2b}$ -COLORING: an FPT algorithm using iterative compression	96
3.4.2 VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING: an FPT algorithm using iterative compression	100
3.4.3 Solving DISJOINT VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING	102
3.4.4 Solving ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING	103
3.4.5 Separators: definitions and notation	104
3.4.6 Solving CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING	105
3.4.7 Solving CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING	107
3.4.7.1 Proof of Lemma 3.35	111
3.4.8 Complexity analysis	114

3.5	Edge Deletion Signed-(H, π)-Coloring	115
3.6	Conclusion and perspectives	117

3.1 Preliminaries

3.1.1 Some known complexity dichotomies

For a t -edge-colored graph H , recall that whenever H -COLORING is NP-complete, VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING are NP-complete even for $k = 0$, and thus are not in XP, unless $P = NP$. For example, this is the case when H is a monochromatic triangle.

On the other hand, when H -COLORING is in P, both problems are in XP for parameter k by a brute-force algorithm iterating over all k -subsets of vertices (resp. edges) of G , performing the operation on these k vertices (resp. edges), and then solving H -COLORING.

Similarly, for a signed graph (H, π) , LIMITED SWITCHINGS (H, π) -COLORING is NP-complete even for $k = 0$, if (H, π) -COLORING is NP-complete. If (H, π) -COLORING is in P, then LIMITED SWITCHINGS (H, π) -COLORING is in XP, by the same brute-force argument.

When SIGNED- (H, π) -COLORING is NP-complete, then LIMITED SWITCHINGS (H, π) -COLORING is NP-complete (but could still be in XP or FPT), and VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING are NP-complete, even for $k = 0$. Moreover if SIGNED- (H, π) -COLORING is in P, then VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING are in XP.

The previous arguments highlight that to determine the complexity of our five problems, we must first know the complexity of H -COLORING for any edge-colored graphs H , and the complexity of (H, π) -COLORING and SIGNED- (H, π) -COLORING for any signed graph (H, π) .

When $k = 0$ and H is 1-edge-colored (*i.e.* H is an undirected graph), we have the following classic theorem.

Theorem 3.1 (Hell and Nešetřil [98]). *Let H be a 1-edge-colored graph. The problem H -COLORING is in P if the core of H has at most one edge (H is bipartite or has a loop), and NP-complete otherwise.*

There is no analogue of Theorem 3.1 for edge-colored graphs. In fact, it is proved in [28] that a dichotomy classification for H -COLORING restricted to 2-edge-colored H would imply a dichotomy for all fixed-target CSP problems. Thus, no simple combinatorial classification is expected to exist. In fact, even for trees, cycles or complete graphs, such classifications are far from trivial, see the PhD thesis [26] for an overview of some partial results highlighting the difficulty of the problem. Some classifications exist for certain classes of graphs H , such as those of order at most 2 (see [16] and [27]) or paths [25]. By isomorphism between signed graphs and 2-edge-colored graphs, the problem (H, π) -COLORING is also unlikely to have such a dichotomy.

Hence for the three problems VERTEX DELETION H -COLORING, EDGE DELETION H -COLORING and LIMITED SWITCHINGS (H, π) -COLORING we focus most of our attention on targets of order at most 2 since H -COLORING and (H, π) -COLORING are polynomial for them (see [16, 27] or Theorem 3.18).

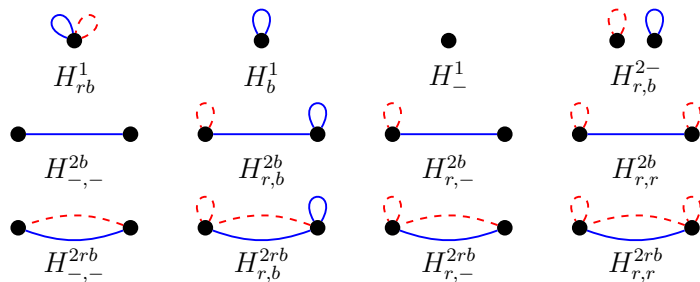


Figure 3.1: The twelve 2-edge-colored cores of order at most 2 considered in this chapter.

The twelve 2-edge-colored graphs of order at most 2 that are cores (up to symmetries of the colors) are depicted in Figure 3.1. The two depicted colors are red (dashed edges) and blue (solid edges). We use the terminology of [16]: for $\alpha \in \{-, r, b, rb\}$, the 2-edge-colored graph H_α^1 is the graph of order 1 with no loop, a red loop, a blue loop, and both kinds of loops, respectively. Similarly, for $\alpha \in \{-, r, b, rb\}$ and $\beta, \gamma \in \{-, r, b\}$, the graph $H_{\beta,\gamma}^{2\alpha}$ denotes the graph of order 2 with vertex set $\{0, 1\}$. The string α indicates the presence of an edge between 0 and 1: no edge, a red edge, a blue edge and both edges for $-$, r , b and rb , respectively. Similarly, β and γ denote the presence of a loop at vertices 0 and 1, respectively ($-$ for no loop, r for a red loop, b for a blue loop).

When working on the context of signed graphs (*i.e.* for LIMITED SWITCHINGS (H, π) -COLORING, VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING), we use the notation $H_{\beta,\gamma}^{2\alpha}$ to refer to the signed graph obtained from the 2-edge-colored graph $H_{\beta,\gamma}^{2\alpha}$ by making the red edges negative and the blue edges positive.

For SIGNED- (H, π) -COLORING, we recall Theorem 2.22.

Theorem 2.22 (Brewster, Foucaud, Hell and Naserasr [28] and Brewster and Siggers [29]). *Let (H, π) be a signed graph. The problem SIGNED- (H, π) -COLORING is in P if the s-core of (H, π) has at most two edges, and is NP-complete otherwise.*

Note that signed graphs where the s-core has at most two edges either have one vertex (with zero loop, one loop or two loops of different signs), or two vertices (with either one edge or two parallel edges of different signs joining them) [28]. If there are two vertices joined by one edge and a loop at one of the vertices, we can switch at the non-loop vertex if necessary to obtain a signed graph with only positive or only negative edges, and then retract the whole graph to the loop-vertex, so this is not an s-core.

By Theorem 2.22, the signed graphs (H, π) for which SIGNED- (H, π) -COLORING is polynomial are H_-^1 (\bullet), H_{rb}^1 ($\bullet \text{---} \bullet$), H_b^1 ($\bullet \text{---} \bullet$), H_r^1 ($\bullet \text{---} \bullet$), $H_{-,-}^{2b}$ ($\bullet \text{---} \bullet$), $H_{-,-}^{2rb}$ ($\bullet \text{---} \bullet$) and $H_{r,b}^{2-}$ ($\bullet \text{---} \bullet$).

3.1.2 Homomorphism dualities and FPT time

For a t -edge-colored graph H , we say that H has the *duality property* if there is a set $\mathcal{F}(H)$ of t -edge-colored graphs such that, for any t -edge-colored graph G , $G \rightarrow H$ if and only if no graph F of $\mathcal{F}(H)$ satisfies $F \rightarrow G$. If $\mathcal{F}(H)$ is finite, we say that H has the *finite duality property*. If checking whether any graph F in $\mathcal{F}(H)$ satisfies $F \rightarrow G$ (for an input edge-colored graph G) is in P, we say that H has the *polynomial duality*

property. This is in particular the case when $\mathcal{F}(H)$ is finite. For such H , H -COLORING is in P. This topic is explored in detail for edge-colored graphs in [16]. By a simple bounded search tree argument, we get the following:

Proposition 3.2. *Let H be a t -edge-colored graph with the finite duality property. Let $c = \max\{|V(F)|, F \in \mathcal{F}(H)\}$.*

The problem VERTEX DELETION H -COLORING can be solved in time $O(f(\mathcal{F}(H))n^c)$ for some computable function f .

The problem EDGE DELETION H -COLORING can be solved in time $O(f(\mathcal{F}(H))n^{c^2})$ for some computable function f .


If $t = 2$ and (H, π) is the signed graph isomorphic to H , then LIMITED SWITCHINGS (H, π) -COLORING can be solved in time $O(f(\mathcal{F}(H))n^c)$ for some computable function f .

Proof. First, we search for all occurrences of homomorphic images of graphs in $\mathcal{F}(H)$ (there are at most $f(\mathcal{F}(H))$ such images for some exponential function f), which we call *obstructions*. This takes time at most $O(f(\mathcal{F}(H))n^c)$, where $c = \max\{|V(F)|, F \in \mathcal{F}(H)\}$. Then, we need to get rid of each obstruction. For VERTEX DELETION H -COLORING (resp. EDGE DELETION H -COLORING), we need to delete at least one vertex (resp. edge) in each obstruction, thus we can branch on all c (resp. c^2) possibilities. For LIMITED SWITCHINGS (H, π) -COLORING, we need to switch at least one of the vertices of the obstruction (but then update the list of obstructions, as we may have created a new one). In all cases, this gives a search tree of height k and degree bounded by a function of $\mathcal{F}(H)$, which is FPT. \square

Some dualities have been obtained for small edge-colored graphs. The following theorem from [16] is crucial for our techniques.

Theorem 3.3 (Bawar, Brewster and Marcotte [16]). *Let H be an edge-colored graph of order at most 2. Then, H has the polynomial duality property. If H has order 1, then H has the finite duality property.*


We next describe the duality sets for some special cases that will be used in our proofs.

Lemma 3.4 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,r}^{2b}$ () if and only if it contains no homomorphic image of cycles with an odd number of blue edges.*


We present a brief proof of their result. Note that homomorphic images of paths are walks and that homomorphic images of cycles are closed walks.

Proof. Let G be a 2-edge-colored graph which admits a homomorphism φ to $H_{r,r}^{2b}$. Suppose that G contains a homomorphic image of some cycle with an odd number of blue edges, that is to say G contains a closed walk W with an odd number of blue edges. Note that if uv is a blue edge, then $\varphi(u) \neq \varphi(v)$ and if uv is a red edge, then $\varphi(u) = \varphi(v)$. By going around the closed walk, we obtain $\varphi(u) \neq \varphi(u)$ for any vertex u of W , a contradiction.


Let G be a 2-edge-colored graph which contains no homomorphic image of cycles with an odd number of blue edges. We identify every connected red components of G . The graph that we obtain has red loops but no other red edges, moreover the graph induced by the blue components is bipartite (otherwise there would be a cycle with an odd number of blue edges in G). Hence by identifying the vertices of each bipartition, we obtain $H_{r,r}^{2b}$. Hence $G \rightarrow H_{r,r}^{2b}$. \square

Lemma 3.5 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,b}^{2b}$ () if and only if it contains no homomorphic image of a red-blue-red 4-vertex path.*

Proof. Let u be the vertex of $H_{r,b}^{2b}$ with a red loop, and v the vertex with a blue loop. Given a 2-edge-colored graph G , map all the vertices incident with a red edge to u , and map all others to v . This is a homomorphism unless two vertices mapped to u are joined by a blue edge. But in this case, we can find a homomorphic image of a red-blue-red walk in G . Conversely, note that a red-blue-red path has no homomorphism to $H_{r,b}^{2b}$. \square


Lemma 3.6 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,-}^{2b}$ () if and only if it contains no homomorphic image of a path of the form $RB^{2p-1}R$ (where R is a red edge, B a blue edge and $p \geq 1$ is an integer) or of cycles with an odd number of blue edges.*


Proof (sketch). First note that none of the two obstructions admit a homomorphism to $H_{r,-}^{2b}$. If a 2-edge-colored graph G has none of these homomorphic images then by identifying every vertex incident with a red edge of G , we obtain a bipartite graph on the blue edges for which one of the two partitions contains every vertex incident with a red loop. By mapping this partition to the vertex of $H_{r,-}^{2b}$ with the red loop and the other partition to the other vertex, we obtain our homomorphism. \square

Lemma 3.7 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,r}^{2rb}$ () if and only if it contains no homomorphic image of an all blue odd cycle.*

Proof (sketch). The idea is to note that the graph induced by the blue edges is bipartite and that the red edges does not create any constraints. \square

The proof of the following results are more complicated, hence we refer the reader to [16] for the details. In a 2-edge-colored graph, a closed walk $v_0v_1 \dots v_t$ is *alternating* if for every $i < t$, v_iv_{i+1} and $v_{i+1}v_{i+2}$ do not have the same color (where the indices are taken modulo t). An alternating closed walk in a 2-edge-colored graph correspond exactly to the notion of alternating closed walk in the isomorphic signed graph. An *odd figure eight* is a closed walk of the form $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$, *i.e.* two odd cycles which share a vertex v_0 .

Lemma 3.8 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,-}^{2rb}$ () if and only if it contains no homomorphic image of an odd figure eight $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$ for which all edges $v_{2i}v_{2i+1}$ are blue.*

Lemma 3.9 (Bawar, Brewster and Marcotte [16]). *A 2-edge-colored graph has a homomorphism to $H_{r,b}^{2rb}$ () if and only if it contains no homomorphic image of alternating odd figure eight, that is, an alternating closed walk $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$.*

3.1.3 Reformulating some modification problems

As mentioned in the introduction, behind the generality of our modification problems lies some interesting particular cases. This section is dedicated to highlighting well known problems which are captured by our five general problems.


The first problem is VERTEX COVER which consists in finding the smallest set S of vertices so that every edge of the input graph is incident with a vertex of S . This problem is equivalent to the two problems VERTEX DELETION H_-^1 -COLORING and VERTEX DELETION SIGNED- H_-^1 -COLORING. In a similar fashion, VERTEX DELETION H_b^1 -COLORING is equivalent to solving VERTEX COVER on the input graph where the blue edges are removed.

An undirected graph G admits a homomorphism to K_2 if and only if it is bipartite. Consequently, the problem ODD CYCLE TRANSVERSAL (resp. EDGE BIPARTIZATION) which consists in finding the minimum number of vertices (resp. edges) to remove to make the graph bipartite, is equivalent to VERTEX DELETION K_2 -COLORING (resp. EDGE DELETION K_2 -COLORING). Note that this problem is also equivalent to VERTEX DELETION H -COLORING (resp. EDGE DELETION H -COLORING) where H is the t -edge-colored graph on two vertices u and v such that for every edge-color i , there is an edge uv colored i . It is also equivalent to VERTEX DELETION SIGNED- $H_{-, -}^{2rb}$ -COLORING (resp. EDGE DELETION SIGNED- $H_{-, -}^{2rb}$ -COLORING) for signed graphs.

It is also possible to encompass a combination of problems. Solving VERTEX DELETION $H_{-, -}^{2b}$ -COLORING is equivalent to solving ODD CYCLE TRANSVERSAL and VERTEX COVER on the input graph where the blue edges are removed at the same time.

Our problems can also easily encode stronger versions of well known problems. For example consider the following problem.

ANNOTATED ODD CYCLE TRANSVERSAL **Parameter:** k
Input: A graph G , two sets A_0 and B_0 of vertices of G and an integer k .
Question: Is there a subset S of vertices of G such that $G - S$ is bipartite with bipartition (A, B) such that $A_0 \subseteq A$ and $B_0 \subseteq B$?

This problem can be seen as a subproblem of VERTEX DELETION H -COLORING where H is the 3-edge-colored graph . Indeed, one can “mark” the two bipartitions with pendant red or green edges (depending on which set among A_0 or B_0 the vertex belongs to) and perform ODD CYCLE TRANSVERSAL on the blue graph. This problem is not the only one, for example we can perform the same operation for the edge deletion version. In general, we can even encode list coloring with homomorphisms of edge-colored graphs: each subset of colors is associated with its own edge-color. The target graph is a blue K_p (for some integer p) where each vertex is incident to loops of every edge-color associated with a list containing this vertex. It suffices to add a loop for each vertex u of the input graph with edge-color corresponding with the list of u .

For signed graphs, many interesting problems can be formulated as homomorphism problems. For example SIGNED- H_b^1 -COLORING is equivalent to determining if the input signed graph is balanced or not. Equivalently SIGNED- H_r^1 -COLORING tests whether the input signed graph is antibalanced or not. Hence the problem VERTEX DELETION SIGNED- H_b^1 -COLORING consists in finding the minimum number of vertices to remove to make the graph balanced. The problem EDGE DELETION SIGNED- H_b^1 -COLORING is equivalent to computing the frustration index of the input signed graph, a problem which has been extensively studied (see Section 2.1.4). The problem LIMITED SWITCHINGS H_b^1 -COLORING consists in not only determining whether the input signed graph is balanced but also the number of switchings necessary to make the input all-positive.

The problems VERTEX/EDGE DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING consist in removing vertices/edges in order for the input signed graph to be both balanced and bipartite,

that is, we want to have chromatic number at most 2. This can be reformulated as wanting to remove both odd cycles and unbalanced cycles.

Unfortunately, in general, determining the chromatic number of a signed graph cannot be expressed as some SIGNED- (H, π) -COLORING problem due to the fact that there exist multiple targets of the same order which are not equivalent.

3.2 Edge-colored modification problems

In this section, we focus on the complexity of the two problems: VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING.

We first adapt a general method from [141] to show that VERTEX DELETION H -COLORING is either trivial, or NP-complete in Section 3.2.1.

For EDGE DELETION H -COLORING, we cannot use this technique (in fact there exist non-trivial polynomial cases). Thus, we turn our attention to edge-colored graphs of order 2 (note that for every edge-colored graph H of order at most 2, H -COLORING is in P [16, 27]). In Section 3.2.2, we prove a dichotomy result for graphs of order at most 2 for the EDGE DELETION H -COLORING problem.

Finally, in Section 3.2.3, we provide FPT algorithms from VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING when H has order 2.

3.2.1 Vertex Deletion H -Coloring: P/NP-complete dichotomy

Graph modification problems for operations vertex-deletion and edge-deletion have been studied extensively. For a graph property \mathcal{P} , we denote by VERTEX DELETION- \mathcal{P} the graph modification problem for property \mathcal{P} and operation vertex-deletion. A property is *hereditary* if $\mathcal{P}(G)$ implies $\mathcal{P}(H)$ for all induced subgraphs H of G . Lewis and Yannakakis [141] defined a non-trivial property \mathcal{P} on graphs as a property true for infinitely many graphs and false for infinitely many graphs. These definitions can be extended to (m, n) -mixed multi-graphs (which contains edge-colored graphs). They showed the following general result.

Theorem 3.10 (Lewis and Yannakakis [141]). *The VERTEX DELETION- \mathcal{P} problem for non-trivial graph-properties \mathcal{P} that are hereditary is NP-hard.*

By modifying the proof of Theorem 3.10, we can prove the two following results.

Theorem 3.11. *Let \mathcal{P} be a non-trivial property of (m, n) -mixed multi-graphs that is hereditary and true for all empty graphs. For such a property, the problem VERTEX DELETION- \mathcal{P} is NP-hard.*

The proof of this theorem follows the proof of Theorem 3.10 from [141]. The only difference is that we work with (m, n) -mixed multi-graphs instead of undirected graphs.

Proof. Let G be an (m, n) -mixed multi-graph. We denote by $CC(G)$ the set of connected components of G . These components are also (m, n) -mixed multi-graphs. For x and v two vertices of G , let $R_v(x)$ be the set of vertices connected to x in $G - v$. For any vertex $v \in V(G)$, let $CC_v(G)$ be the set of connected subgraphs of G induced by the sets of vertices of the form $R_v(x) \cup \{x\}$ for $x \in V(G - v)$. In other words, $CC_v(G)$ is the set of

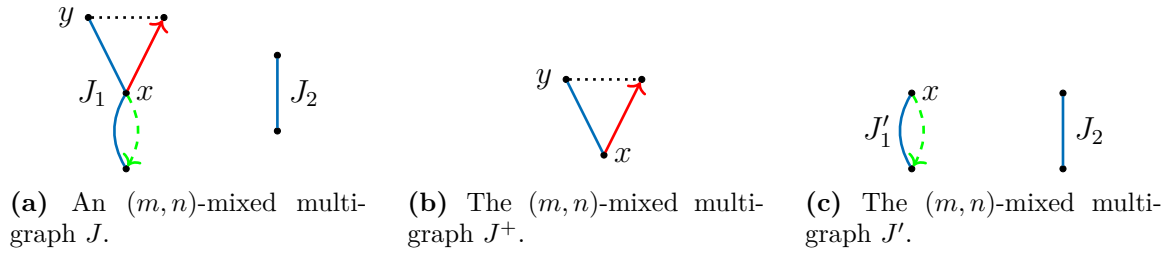


Figure 3.2: An example of a $(2, 2)$ -mixed multi-graph J and its induced subgraphs J_1 , J_2 , J^+ , J'_1 and J' .

connected components of $G - v$ where we added the vertex v . In particular, if v is not a cut-vertex, then $CC_v(G) = \{G\}$.

For a connected (m, n) -mixed multi-graph G and $v \in G$, let $\alpha_v(G) = (n_1, n_2, \dots, n_t)$ such that $n_1 \geq n_2 \geq \dots \geq n_t$, and the multi-sets $\{n_1, \dots, n_t\}$ and $\{|V(C)| : C \in CC_v(G)\}$ are equal. In other words, $\alpha_v(G)$ is the ordered sequence of the orders of the (m, n) -mixed multi-graphs in $CC_v(G)$. Let $\alpha(G)$ be the smallest sequence (for the lexicographic order) $\alpha_v(G)$ over all possible vertices $v \in V(G)$.

For an (m, n) -mixed multi-graph G , let $\beta(G) = (\alpha(G_1), \alpha(G_1), \dots, \alpha(G_t))$ such that $\alpha(G_1) \geq_L \alpha(G_2) \geq_L \dots \geq_L \alpha(G_t)$ (where \geq_L is the lexicographical order) and $CC(G) = \{G_1, \dots, G_t\}$. In other words, $\beta(G)$ is the ordered sequence of α -sequences of the connected components of G .

Recall that \mathcal{P} is non-trivial. In particular, \mathcal{P} has counter-examples. For an integer p and an (m, n) -mixed multi-graph G , we denote by pG , the (m, n) -mixed multi-graph composed of p disjoint copies of G . Let J be an (m, n) -mixed multi-graph such there exists some $k \geq 1$ for which $\mathcal{P}(kJ)$ is false, and which has the minimum β -sequence among the (m, n) -mixed multi-graphs verifying this property. Let $k \geq 1$ such that $\mathcal{P}(kJ)$ is false and $\mathcal{P}((k-1)J)$ is true. Suppose that $\beta(J) = (\alpha(J_1), \dots, \alpha(J_t))$ where $CC(J) = \{J_1, \dots, J_t\}$. Let x be a vertex of J_1 for which $\alpha(J_1) = \alpha_x(J_1)$ and let J^+ be the connected (m, n) -mixed multi-graph of $CC_x(J_1)$ with the greatest number of vertices. Since all empty graphs verify \mathcal{P} , J contains at least one edge. This implies that J_1 and J^+ contain at least one edge. In particular, J^+ contains at least two vertices. Let y be a vertex of J^+ which is different from x . Let J'_1 (resp. J') be the (m, n) -mixed multi-graph obtained from J_1 (resp. J) by removing the vertices of $V(J^+) \setminus \{x, y\}$. See Figure 3.2 for an example.

Each induced subgraph of J that we defined will be useful to show that VERTEX DELETION- \mathcal{P} is NP-hard. We reduce VERTEX COVER to VERTEX DELETION- \mathcal{P} . Note that it may be complicated to find the (m, n) -mixed multi-graph J . If we cannot find J in polynomial time then it makes the reduction non-constructive. Let (G, ℓ) be an instance of VERTEX COVER where G is an undirected graph of order p and ℓ is an integer.

We construct the (m, n) -mixed multi-graph H from G as follows. For each vertex $v \in V(G)$, we add a copy J'_v of J' to H . For each edge $uv \in E(G)$, we add a copy J_{uv}^+ of J^+ to H . We identify the copy x_u (resp. x_v) of x in J'_u (resp. J'_v) with the copy x_{uv} (resp. y_{uv}) of x (resp. y) in J_{uv}^+ . This ends the construction of H . See Figure 3.3 for an example. We construct the (m, n) -mixed multi-graph H' by taking pk disjoint copies of H .

We claim that (G, ℓ) is a positive instance of VERTEX COVER if and only if $(H', pk\ell)$ is a positive instance of VERTEX DELETION- \mathcal{P} .

Suppose that there is a subset S of vertices of G of size at most ℓ that is a vertex

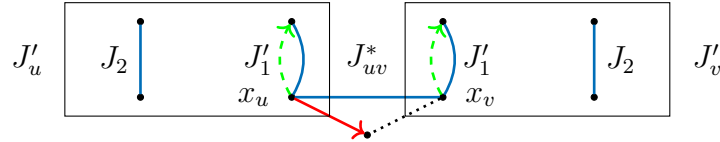


Figure 3.3: An example of the graph H when J is the $(2, 2)$ -mixed multi-graph of Figure 3.2 and G is just an edge uv . Here, we chose to identify x_u with x_{uv} and x_v with y_{uv} . Note that if no vertex is removed from H , then H contains J as an induced subgraph.

cover of G . We construct $S' \subseteq V(H')$ as follows. For every copy of H in H' and every vertex $u \in S$, we add the copy of the vertex x_u of J'_u to S' . Note that $|S'| \leq pk\ell$. We claim that $H' - S'$ verifies \mathcal{P} . Let \mathcal{J} be the set of (m, n) -mixed multi-graphs that can be constructed as follows. Take a copy of J'_1 and at most $\Delta(G)$ copies of J^+ . For each copy of J^+ , delete one of x or y and identify the other vertex with the copy x' of x in the copy of J'_1 . The set \mathcal{J} contains at most $3^{\Delta(G)}$ (m, n) -mixed multi-graphs, \mathcal{J} contains all possible maximal connected induced subgraph of H connected to a vertex x_u when every x_v for $v \in N(u)$ has been removed in H .

A connected component C of $H' - S'$ can be one of the following four types:

1. The connected component C belongs to $\{J_2, \dots, J_t\}$.
2. The connected component C belongs to \mathcal{J} .
3. The connected component C is isomorphic to a connected induced subgraph of J'_1 where the vertex x has been removed.
4. The connected component C is isomorphic to a connected induced subgraph of J^+ where the vertices x and y have been removed.

Let J^+ be the (m, n) -mixed multi-graph composed of disjoint copies of the vertices of \mathcal{J} and disjoint copies of J_2, \dots, J_{t-1} and J_t . Note that every connected component of $H' - S'$ is an induced subgraph of J^* . Let $C \in \mathcal{J}$, note that $\alpha(C) \leq \alpha_{x'}(C)$ where x' is the copy of x in J'_1 . Note that $CC_{x'}(C) = CC_{x'}(J') \cup X$ where X is the set corresponding to the copies of J^+ in C with one of x or y removed. The connected multi-graphs of X have order $|V(J^+)| - 1$, hence $\alpha_{x'}(C) <_L \alpha_x(J_1) = \alpha(J_1)$. Note that $\beta(J^*) <_L \beta(J)$ since for every $C \in \mathcal{J}$, $\alpha(C) <_L \alpha(J_1)$.

By minimality of J , any number of disjoint copies of J^* must verify \mathcal{P} , hence $H' - S'$ verifies \mathcal{P} and $(H', pk\ell)$ is a positive instance of VERTEX DELETION- \mathcal{P} .

Suppose that there is a subset S' of vertices of H' of size at most $pk\ell$ such that $\mathcal{P}(H' - S')$ holds. Note that $H' - S'$ can contain at most $k - 1$ copies of the (m, n) -mixed multi-graph J by definition of J . In particular H' has at least $pk - (k - 1)$ copies of H for which after removing the vertices of S' , the (m, n) -mixed multi-graph does not contain a copy of J .

Suppose that for one of the copies H_0 of H , $|V(H_0) \cap S'| \leq \ell$. In this case, we construct $S \subseteq V(G)$ as follows. If $S' \cap V(J'_u) \neq \emptyset$, then add u to S . If $S' \cap (V(J'_{uv}) \setminus \{x, y\}) \neq \emptyset$, then add arbitrarily one of u or v to S . Note that $|S| \leq \ell$. Suppose that there is an edge $uv \in E(G)$, such that $u, v \notin S$. Our copy of H contains J'_u, J'_v and J'_{uv} and these (m, n) -mixed multi-graphs do not contain vertices from S . The vertex x_{uv} has been identified with one of x_u or x_v , say x_u . The (m, n) -mixed multi-graph composed of J'_u and J'_{uv} with x_u and x_{uv} identified is exactly the (m, n) -mixed multi-graph J . Hence if $H - S'$ does not contain J , then the set S is a vertex cover of G of size at most ℓ .

Suppose, by contradiction, that for every copy of H either $H - S'$ contains J or verifies $|V(H_0) \cap S'| \geq \ell + 1$. In this case, S' has at least $(pk - (k - 1))(\ell + 1)$ vertices. Moreover, as $\ell < p$ (otherwise the instance of VERTEX COVER is trivial), $(pk - (k - 1))(\ell + 1) \geq pk\ell + \ell + 1 + k(p - (\ell - 1)) > pk\ell$, a contradiction.

Hence G has a vertex cover of size at most ℓ . \square

For a t -edge-colored graph, the only case where the property of mapping to H is trivial (in this case, always true) is when H has a vertex with a loop of each edge-color (in which case the core of H is this vertex). Thus we obtain the following dichotomy.

Corollary 3.12. *Let H be a t -edge-colored graph. VERTEX DELETION H -COLORING is in P if H contains a vertex having a loop of each edge-color, and NP-complete otherwise.*

Proof. For every t -edge-colored graph H , VERTEX DELETION H -COLORING is in NP. For a t -edge-colored graph G , the property $\mathcal{P}(G) : "G \rightarrow H"$ is an hereditary property and is verified by all independent sets, thus if it has infinitely many negative instances (on loopless t -edge-colored graphs), then it is non-trivial, and thus NP-hard. Let us see when this is the case.

We can observe that the problem is actually trivial if H contains a vertex with all t -colored loops, indeed every t -edge-colored graph can be mapped to this vertex (in this case, we accept). Moreover, if not, then the complete graph $K_{|V(H)|+1}$ with all t -colored edges between each pair of vertices does not map to H . Indeed by the pigeonhole principle, two vertices u and v of our input t -edge-colored graph must have the same image vertex w in H . As there is an edge colored i between u and v , there must be a loop colored i on w . Thus w should have all t kinds of loops, a contradiction. Thus, in all such cases, the property is non-trivial on loopless t -edge-colored graphs and hence the problem is NP-complete. \square

3.2.2 Edge Deletion H -Coloring: P/NP-complete dichotomy when H has order 2

No analogue of Theorem 3.10 for the operation edge-deletion exists nor is expected to exist [205]. We thus restrict our attention to the case of edge-colored graphs H of order at most 2. For this case we classify the complexity of EDGE DELETION H -COLORING. Since multiple edges of the same color are irrelevant, if H has order 2, for each edge-color there are three possible edges.

Theorem 3.13. *Let H be an edge-colored core of order at most 2. If each color class of the edges of H contains either only loops or all three possible edges, then EDGE DELETION H -COLORING is in P; otherwise it is NP-complete.*

We separate the proof of this theorem into several lemmas.

Lemma 3.14. *Let H be an edge-colored core of order at most 2. If each color class of the edges of H contains either only loops or all three possible edges, then EDGE DELETION H -COLORING is in P.*

Proof. First note that if color i has all three possible edges in H , we can simply ignore this color by removing it from H and G without decreasing the parameter, as it does not provide any constraint on the homomorphism.

We can therefore suppose that H contains only loops. If two colors induce the same subgraph of H , then we can identify these two colors in both G and H as they give the same constraints.

If G has colors that H does not have, then remove each edge with this color and decrease the parameter for each removed edge. If it goes below zero then we reject.


We can now assume that H has only loops and G has the same colors as H . We are left with only three cases, as H is a core (there is no vertex whose set of loops is included in the set of loops of the other).

1. H has a single loop. Then, $G \rightarrow H$ as G has the same colors as H .
2. H has one loop colored a and one loop colored c on the first vertex and has one loop colored b and one loop colored c on the second vertex. Up to symmetry, suppose that H has one blue loop and one green loop on the first vertex and has one red loop and one green loop on the second vertex. We will reduce to the problem where we have removed the green loops. Let p be the number of green edges of G . We construct G' from G by replacing each green edge by a blue edge and a red edge (we can end up with multiple blue or red edges that way). We claim that EDGE DELETION H -COLORING with parameter k and input G is true if and only if EDGE DELETION $H_{r,b}^{2-}$ -COLORING with parameter $k + p$ on input G' is true.

If the first problem has a solution S , then remove the corresponding edges from G' (if the original edge of G is green remove the two new edges in G'). Each vertex of $G - S$ is set to one component, in particular each green edge is set to a vertex with a blue edge or a red edge. If a green edge uv of G is sent to the first vertex (resp. second vertex), we remove the edge of G' corresponding to uv which is red (resp. blue). We can check that after removing those edges, G' admits a homomorphism to $H_{r,b}^{2-}$. We removed at most k edges in the first step plus the number of green edges in S and removed one edge for each green edge left in the second step. Thus, we removed at most $k + p$ edges in G' .

If the second problem has a solution S , then remove from G all blue and red edges of S . Remove the green edges of G only if both were removed in G' . Note that S contains at least one edge in G' for each green edge of G . Thus we removed at most k edges in G . Moreover, $G \rightarrow H$ by taking the same homomorphism as in G' . Indeed, the blue and red edges are sent to one of the two loops while each green connected component is sent to one vertex.

Using this method we can reduce the problem to EDGE DELETION $H_{r,b}^{2-}$ -COLORING, which is our last case.

3. H contains two non-incident loops with different colors. In this case, $H = H_{r,b}^{2-}$ (). Indeed if there were any other kind of loop, then we would be in the previous case or we could identify two colors. Note that a 2-edge-colored graph maps to $H_{r,b}^{2-}$ if and only if it has no red edge incident to a blue edge. Thus, solving EDGE DELETION $H_{r,b}^{2-}$ -COLORING amounts to disconnecting red and blue connected components. This can be done by constructing the following bipartite graph: put a vertex for each edge of G ; two vertices are adjacent if the corresponding edges in G are adjacent and of different colors. Solving EDGE DELETION $H_{r,b}^{2-}$ -COLORING is the same as solving VERTEX COVER on this bipartite graph, which is in P.

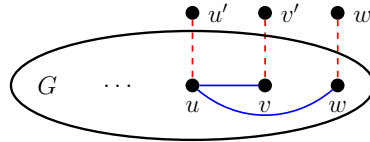


Figure 3.4: Reduction from VERTEX COVER to EDGE DELETION $H_{r,b}^{2b}$ -COLORING.

There is no other case as otherwise the set of loops at one vertex would be included in the set of loops at the other. \square

The NP-completeness proofs are by reductions from VERTEX COVER, based on vertex- and edge-gadgets constructed using obstructions to the corresponding homomorphisms from [16].

We start with proving the NP-hardness of two special cases, and then we will show that we can always reduce the problem from these two cases.

Lemma 3.15. *The problem EDGE DELETION $H_{r,b}^{2b}$ -COLORING (i.e. ) is NP-hard.*

Proof. We reduce from VERTEX COVER. Given an input graph G of VERTEX COVER, we construct a 2-edge-colored graph G' from G as follows. Take G and color all edges blue, then add a pending red edge vv' to each vertex v of G (see Figure 3.4). By Lemma 3.5, a 2-edge-colored graph maps to $H_{r,b}^{2b}$ if and only if it does not contain a homomorphic image of a red-blue-red path [16] i.e. a path $v_0v_1v_2v_3$ where v_0v_1 and v_2v_3 are red and v_1v_2 is blue.

Assume that G has a vertex cover C of size at most k . When removing the edges of the form vv' for $v \in C$ in G' , the resulting graph does not contain red-blue-red paths and thus maps to $H_{r,b}^{2b}$.

Conversely, assume that we have a set S of k edges of G' such that $(G' - S) \rightarrow H_{r,b}^{2b}$. In particular, for every blue edge uv of G , we must have one of uu', uv or vv' in S . Thus we can obtain a vertex cover of G of size k from S : for a vertex v , if vv' belongs to S , we add v to that vertex cover. If $uv \in S$, we add either u or v to the vertex cover.

We thus have a polynomial-time reduction from VERTEX COVER to EDGE DELETION $H_{r,b}^{2b}$ -COLORING. Therefore this problem is NP-hard. \square

Lemma 3.16. *The problem EDGE DELETION $H_{r,b}^{2rb}$ -COLORING (i.e. ) is NP-hard.*

Proof. We again reduce from VERTEX COVER. For an input graph G of VERTEX COVER, we construct a 2-edge-colored graph G' from G as follows. We start with a red copy of G , then we add a pending blue edge vv' for each $v \in G$. Finally, for each edge $uv \in G$, we create three new vertices x_{uv}, y_{uv}, z_{uv} such that $u'x_{uv}, v'x_{uv}, y_{uv}z_{uv}$ are red and $x_{uv}y_{uv}, x_{uv}z_{uv}$ are blue (see Figure 3.5).

We then recall Lemma 3.9 proved in [16], stating that a 2-edge-colored graph maps to $H_{r,b}^{2rb}$ if and only if it does not contain an *alternating odd figure eight*, that is, an alternating closed walk $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$. Note that our construction creates such a pattern for each edge of G .

Assume that G has a vertex cover C of size at most k . Then for each $v \in C$, we delete vv' from G' . We prove that the resulting graph G'' contains no alternating odd figure

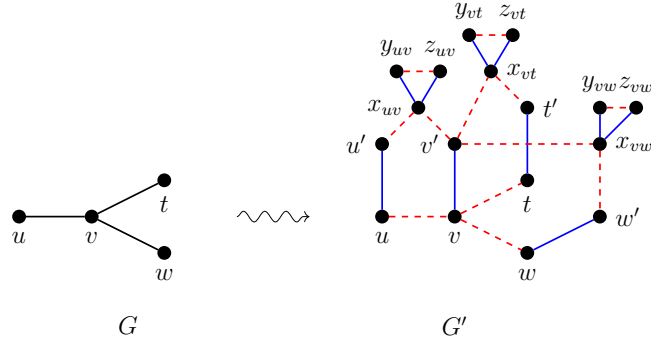


Figure 3.5: Reduction from VERTEX COVER to EDGE DELETION $H_{r,b}^{2rb}$ -COLORING


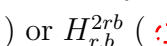
eight. First observe that in the graph obtained from G' by removing all edges from G , all the alternating walks have length at most 7, hence it contains no alternating odd figure eight. Thus, if G'' contains an alternating odd figure eight, then it uses an edge uv from G . Since C is a vertex cover, either uu' or vv' is not present in G'' . Then, either u or v has no incident blue edge. This implies that G'' has no alternating odd figure eight, and hence maps to $H_{r,b}^{2rb}$.

Conversely, assume that we can remove a set S of k edges from G' so that $G' \setminus S \rightarrow H_{r,b}^{2rb}$. We construct a set $C \subset V(G)$ as follows: if $vv' \in S$, then we add $v \in C$. If $uv, u'x_{uv}, v'x_{uv}, x_{uv}y_{uv}, x_{uv}z_{uv}$ or $y_{uv}z_{uv}$ lie in S , then we add either u or v to C . Note that, in each case, $|C| \leq k$. Moreover, we claim that C is a vertex cover of G . Assume not, and consider an edge uv in G such that $u, v \notin C$. By construction, this means that none of the edges $uv, uu', vv', u'x_{uv}, v'x_{uv}, x_{uv}y_{uv}, x_{uv}z_{uv}, y_{uv}z_{uv}$ lies in S . These vertices form an alternating odd figure eight, contradicting that $G' \setminus S \rightarrow H_{r,b}^{2rb}$.

Therefore, EDGE DELETION $H_{r,b}^{2rb}$ -COLORING is NP-hard. □

Lemma 3.17. *For H an edge-colored core of order at most 2, if there exists a color of H which contains an edge which is not a loop and does not contain all three possible edges, then EDGE DELETION H -COLORING is NP-complete.*

Proof. Take such a graph H . If one color, say blue, contains only one edge from the first vertex to the second, then for graphs G which are all blue, the problem is equivalent to EDGE BIPARTIZATION, which is NP-complete.

Now, if H contains no such edge, then by assumption it must contain a color, say blue, with a loop and an edge from the first vertex to the second (and no other edge of this color). Let u be the vertex with the loop and v be the other vertex. Since H is a core, H does not map to its subgraph induced by u . If for every edge-color of H there was a loop of this color on u , then H would not be a core. Hence there exists a color, say red, such that there is a red edge in H and u has no loop colored red. Hence, the graph obtained by removing all edges which are neither blue nor red, is either $H_{r,b}^{2b}$ () or $H_{r,b}^{2rb}$ () up to symmetry. Thus, by the previous two lemmas, Lemma 3.15 and Lemma 3.16, the problem is NP-complete using the same reductions on input edge-colored graphs which only have blue and red edges (the edges of H that are neither blue nor red can be ignored). □

$E_i(H)$	Clause	$E_i(H)$	Clause
\emptyset	$(x_u)(\overline{x_u})$	$\{00, 01\}$	$(\overline{x_u} + \overline{x_v})$
$\{00\}$	$(\overline{x_u})(\overline{x_v})$	$\{01, 11\}$	$(x_u + x_v)$
$\{01\}$	$(x_u + x_v)(\overline{x_u} + \overline{x_v})$	$\{00, 11\}$	$(x_u + \overline{x_v})(\overline{x_u} + x_v)$
$\{11\}$	$(x_u)(x_v)$	$\{00, 01, 11\}$	$(x_u + \overline{x_u})$

Table 3.3: Clauses appearing in the 2-SAT formula $F(G)$ of Theorem 3.18 proved in [27], for each edge uv of G colored i . The clauses depend on the edge set of H in color i , described in the rows (where $V(H) = \{0, 1\}$).

3.2.3 Vertex/Edge Deletion H -Coloring: FPT algorithms when H has order 2

For many edge-colored graphs H of order at most 2, we can show that VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING are FPT by giving ad-hoc reductions to VERTEX COVER, ODD CYCLE TRANSVERSAL or a combination of both. However, a more powerful method is to generalize a technique from [27] used to prove that H -COLORING is in P by reduction to 2-SAT (see also [26]):

Theorem 3.18 (Brewster, Dedić, Huard and Queen [27]). *Let H be an edge-colored graph of order at most 2. Then, for each instance G of H -COLORING, there exists a polynomially computable 2-SAT formula $F(G)$ that is satisfiable if and only if $G \rightarrow H$. Thus, H -COLORING is in P .*

Proof (sketch). The formula $F(G)$ from Theorem 3.18 contains a variable x_v for each vertex v of G , and for each edge uv , a set of clauses that depends on H , as described in Table 3.3 (reproduced from [27]). The idea is to see the two vertices of H as “true” (1) and “false” (0), and for each edge uv of a certain color, to express the possible valid assignments of x_u and x_v based on the edges of that color that are present in H . For example, if H has, for color i , a loop at vertex 0 and an edge 01, but no other edge of color i , for each edge uv of G of color i , we add the clause $(\overline{x_u} + \overline{x_v})$ to $F(G)$, indeed the constraint for edge uv is satisfied if at least one of u, v is mapped to 0. \square

We will show how to generalize this idea to VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING. We will need the following parameterized variant of 2-SAT:

VARIABLE DELETION ALMOST 2-SAT **Parameter:** k .
Input: A 2-CNF Boolean formula F , an integer k .
Question: Is there a set of k variables that can be deleted from F (together with the clauses containing them) so that the resulting formula is satisfiable?

VARIABLE DELETION ALMOST 2-SAT and another similar variant, CLAUSE DELETION ALMOST 2-SAT (where instead of k variables, k clauses may be deleted), are known to be FPT: a solution can be found in $O(2^{O(k)}(n+m)^{O(1)})$ time (see [53, Chapter 3.4] and [165]) where n is the number of variables and m is the number of clauses of the formula. We need to introduce a more general variant, that we call GROUP DELETION ALMOST 2-SAT, defined as follows.

GROUP DELETION ALMOST 2-SAT

Parameter: k .

Input: A 2-CNF Boolean formula F , an integer k , and a partition of the clauses of F into groups such that each group has a variable which is present in all of its clauses.

Question: Is there a set of k groups of clauses that can be deleted from F so that the resulting formula is satisfiable?

By a generalization of [53, Exercise 3.21] for CLAUSE DELETION ALMOST 2-SAT, we obtain the following complexity result for GROUP DELETION ALMOST 2-SAT.

Proposition 3.19. *GROUP DELETION ALMOST 2-SAT is FPT and can be solved in $O(2^{O(k)}(n+m)^{O(1)})$ time where n is the number of variable and m is the number of clauses of the formula.*

Proof. We will reduce the problem GROUP DELETION ALMOST 2-SAT to the problem VARIABLE DELETION ALMOST 2-SAT.

Take an instance \mathcal{G} of GROUP DELETION ALMOST 2-SAT with groups g_1, \dots, g_p . We construct an instance \mathcal{V} of VARIABLE DELETION ALMOST 2-SAT as follows. For $i \in [1, p]$, we replace each occurrence of variable x in the clauses of group g_i with a new variable x_i . Moreover, for each variable x and for each i, j , such that $1 \leq i < j \leq p$, we add the two clauses $(\bar{x}_i + x_j)$ and $(x_i + \bar{x}_j)$ to \mathcal{V} (i.e. $x_i = x_j$). The parameter for \mathcal{V} remains k .

Suppose that \mathcal{V} is a positive instance, i.e. that after removing up to k variables, the resulting set of clauses \mathcal{V}' is satisfied by a truth assignment v . For each removed variable x_i , we remove the group of clauses g_i in \mathcal{G} . Note that at most k groups are removed since we removed at most k variables in \mathcal{V} . We have to show that the new set of clauses \mathcal{G}' is satisfiable.

Note that if x_i and x_j are not removed, then v satisfies $(\bar{x}_i + x_j)$ and $(x_i + \bar{x}_j)$, which ensures that $v(x_i) = v(x_j)$. Thus, defining the truth value of x by the value of $v(x_i)$ (for some non-removed x_i) is well-defined. Take a clause $(x + y)$ of \mathcal{G}' , then $(x_i + y_i)$ is a satisfied clause of \mathcal{V}' for some $i \in [1, p]$. By definition of our truth assignment, $(x + y)$ is satisfied, so \mathcal{G}' is satisfiable. Therefore, \mathcal{G} is a positive instance.

Conversely, suppose that we can remove k groups from \mathcal{G} such that the resulting set of clauses \mathcal{G}' is satisfied by v . If we removed the group g_i in the solution, then we remove x_i in \mathcal{V} where x_i is a variable of g_i that appears in each of its clauses. Such a variable exists by definition of \mathcal{G} . This removes all the clauses corresponding to the clauses of the group g_i in \mathcal{V} . Thus, taking the truth assignment that assigns to each x_i the value $v(x)$ satisfies the instance \mathcal{V} . \square

We are now able to prove the following theorem.

Theorem 3.20. *For every edge-colored graph H of order at most 2, VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING are FPT and can be solved in $O(2^{O(k)}n^{O(1)})$ where n is the order of the input edge-colored graph.*

Proof. For an instance G, k of VERTEX DELETION H -COLORING or EDGE DELETION H -COLORING, we consider the formula $F(G)$ from Theorem 3.18 (see Table 3.3). In $F(G)$, to each vertex of G corresponds a variable x_v . Deleting v from G when mapping G to H has the same effect as deleting x_v when satisfying $F(G)$. Thus, this is an FPT reduction from VERTEX DELETION H -COLORING to VARIABLE DELETION ALMOST 2-SAT.

Moreover, each edge uv of G corresponds to one or two clauses of $F(G)$. This naturally defines the groups of GROUP DELETION ALMOST 2-SAT by grouping the clauses

$A_i(H)$	Clause	$A_i(H)$	Clause
\emptyset	$(x_u)(\bar{x}_u)$	$\{01, 10\}$	$(x_u + x_v)(\bar{x}_u + \bar{x}_v)$
$\{00\}$	$(\bar{x}_u + c)(\bar{x}_v + c)(\bar{c})$	$\{01, 11\}$	(x_v)
$\{01\}$	$(\bar{x}_u + c)(x_v + c)(\bar{c})$	$\{10, 11\}$	(x_u)
$\{10\}$	$(\bar{x}_v + c)(x_u + c)(\bar{c})$	$\{00, 01, 10\}$	$(\bar{x}_u + \bar{x}_v)$
$\{11\}$	$(x_u + c)(x_v + c)(\bar{c})$	$\{00, 01, 11\}$	$(\bar{x}_u + x_v)$
$\{00, 01\}$	(\bar{x}_u)	$\{00, 10, 11\}$	$(x_u + \bar{x}_v)$
$\{00, 10\}$	(\bar{x}_v)	$\{01, 10, 11\}$	$(x_u + x_v)$
$\{00, 11\}$	$(x_u + \bar{x}_v)(\bar{x}_u + x_v)$	$\{00, 01, 10, 11\}$	$(x_u + \bar{x}_u)$

Table 3.4: Clauses appearing in the 2-SAT formula $F(G)$ of Theorem 3.21, for each arc uv of the (n, m) -mixed graph G colored i . The clauses depend on the arc set of H in color i , described in the rows (where $V(H) = \{0, 1\}$). The variable c present in this table is unique to each edge uv , *i.e.* we create a new variable for each edge.

corresponding to the same edge. Removing an edge is equivalent to removing its corresponding group. To finish, we have to make sure that we can have one variable common to all the clauses of each group. This is the case in the reduction in [27] for every case except when $E_i(H)$ (the set of edges of color i in H) is just a loop. Assume without loss of generality that the loop is on vertex 1 (the other loop can be treated the same way). Suppose uv has color i in G ; then uv must be mapped to the loop on vertex 1. The original reduction added the clauses $(x_u)(x_v)$; we modify this part and add instead the clauses $(c + x_u)(c + x_v)(\bar{c})$ where c is a new variable. This is now a valid and equivalent instance of GROUP DELETION ALMOST 2-SAT, which is FPT by Proposition 3.19. \square

It is also possible to generalize our two problems to (m, n) -mixed graphs. Note that the same kind of argument can be generalized to mixed graphs to obtain the following result.

Theorem 3.21. *For every (n, m) -mixed graph H of order 2, VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING are FPT.*

Proof. The proof is sensibly the same as for the previous theorem. Let H be the target (n, m) -mixed graph and G be the input (n, m) -mixed graph. Each edge of G is associated with a group of clauses as in Theorem 3.20. The group of clauses for the arcs of G are described in Table 3.4. This creates a 2-SAT formula $F(G)$.

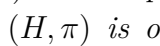
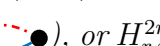
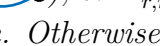


If we want to solve VERTEX DELETION H -COLORING, then we solve VARIABLE DELETION ALMOST 2-SAT on $F(G)$ and k . If we want to solve EDGE DELETION H -COLORING, then we solve GROUP DELETION ALMOST 2-SAT on $F(G)$ and k . Note that in the last case, each group of clauses has a common variable. \square

3.3 Limited Switchings (H, π) -Coloring when H has order 2

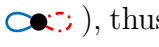


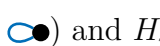


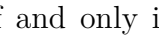


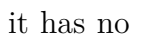
In this section, we study the complexity of the problem LIMITED SWITCHINGS (H, π) -COLORING for signed graphs (H, π) of order at most 2.

3.3.1 Limited Switchings (H, π) -Coloring: P/NP-complete dichotomy

We start by presenting a P/NP-complete dichotomy theorem for each of the sp -cores of order 2. Recall that there is an isomorphism between the set of 2-edge-colored graphs and the set of signed graphs. We use the notations for 2-edge-colored cores when talking about sp -cores (see Figure 3.1).

Theorem 3.22. *Let (H, π) be an sp -core of order at most 2 (i.e. one of the signed graphs in Figure 3.1). If (H, π) is one of $H_{r,b}^{2b}$ () , $H_{r,-}^{2b}$ () , $H_{r,b}^{2rb}$ () , $H_{r,-}^{2rb}$ () , or $H_{r,r}^{2rb}$ () , then LIMITED SWITCHINGS (H, π) -COLORING is NP-complete. Otherwise, it is in P.*

Proof. We begin with the polynomial cases.

- Every signed graph maps to H_{rb}^1 () , thus LIMITED SWITCHINGS H_{rb}^1 -COLORING is trivially in P.
- No graph with an edge can be mapped to H_-^1 () (regardless of switchings).
- For H_b^1 () , we need to test whether the signed graph can be switched to an all-positive graph in less than k switchings or not. There are only two sets of switched vertices that achieve this signature (one is the complement of the other). It is in P to test if the graph can be switched to an all-positive signed graph by Theorem 2.8. Doing that also gives us one of the two sets of switched vertices. We then need to check if its size is at most k or at least $|V(G)| - k$. Hence, LIMITED SWITCHINGS H_b^1 -COLORING is in P.
- For $H_{r,b}^{2-}$ () , we just apply the algorithm for H_b^1 () and H_r^1 () to each connected component, one of the two must accept for each of them.
- For $H_{-, -}^{2rb}$ () , a signed graph (G, σ) is a positive instance if and only if G (without considering edge-colors) is bipartite, which can be tested in polynomial time.
- For $H_{-, -}^{2b}$ () a signed graph (G, σ) is a positive instance if and only if it is bipartite and maps to H_b^1 () . We just need to check the two properties, which are both polynomially testable.
- For $H_{r,r}^{2b}$ () , a signed graph (G, σ) maps to $H_{r,r}^{2b}$ if and only if it has no cycle with an odd number of positive edges (see Lemma 3.4, proved in [16]). This property is preserved under the switching operation. Thus, switching the graph does not impact the nature of the instance. It is thus in P (we can test with $k = 0$) since $H_{r,r}^{2b}$ -COLORING is in P [16, 27].

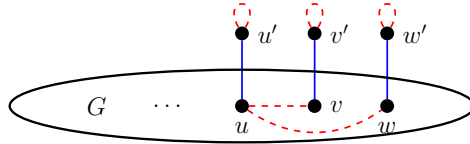







Figure 3.6: Reduction from VERTEX COVER to LIMITED SWITCHINGS $H_{r,-}^{2b}$ -COLORING.

We now consider the NP-complete cases. For every signed graph (H, π) , LIMITED SWITCHINGS (H, π) -COLORING clearly lies in NP. The NP-hardness follows from Theorem 2.22 in all but one case: indeed, $H_{r,b}^{2b}$ (), $H_{r,b}^{2rb}$ (), $H_{r,-}^{2rb}$ (), and $H_{r,r}^{2rb}$ () are their own switching cores and have at least three edges, and thus when (H, π) is one of these, LIMITED SWITCHINGS (H, π) -COLORING is NP-complete, even with $k = |V(G)|$.



The last case is for $H_{r,-}^{2b}$ (). We give a reduction from VERTEX COVER to LIMITED SWITCHINGS $H_{r,-}^{2b}$ -COLORING. Given an instance G, k of VERTEX COVER, we construct the signed graph (G', σ') from an all-negative copy $(G, E(G))$ of G where we attach to each vertex v of G a positive edge vv' , with a negative loop on the new vertex v' (see Figure 3.6).


Denote by x the vertex of $H_{r,-}^{2b}$ with a loop, and by y the other one. Assume that G has a vertex cover C of size at most k . Let (G', σ'') be the signed graph obtained from (G', σ') by switching the vertices of C . We map every vertex of the form v' to x , every vertex of C to x and the remaining ones to y . Since C is a vertex cover, every negative edge of (G', σ'') is either a loop on some vertex v' , an edge vv' with $v \in C$ or an edge uv with $u, v \in C$. In each case, both endpoints are mapped to x . The positive edges of (G', σ'') are then of the form vv' with $v \notin C$ or uv with $u \in C$ and $v \notin C$. In both cases, the two endpoints are mapped to different vertices of $H_{r,-}^{2b}$. Hence, $(G', \sigma'') \xrightarrow{p_s} H_{r,-}^{2b}$ and $G' \xrightarrow{s} H_{r,-}^{2b}$.

Conversely, assume that we can switch the vertices of a set S in (G', σ') such that the resulting signed graph (G', σ'') verifies $(G', \sigma'') \xrightarrow{p_s} H_{r,-}^{2b}$. Let C be the set of vertices v of G such that the vertices v or v' of G' lies in S . Note that C has size at most $|S|$. We claim that C is a vertex cover of G . Assume that there is an edge uv in G with $u, v \notin C$. By construction, $u, u', v, v' \notin S$, so uu', vv' are positive in (G', σ'') , and uv is negative. Thus, u and v have to be mapped to x , and u' and v' have to be mapped to y , a contradiction since u' has an incident negative loop in (G', σ'') . Therefore C is a vertex cover of G . \square


3.3.2 Limited Switchings (H, π) -Coloring: FPT cases

We now consider the parameterized complexity of LIMITED SWITCHINGS (H, π) -COLORING. By Theorem 3.22, there are five signed graphs (H, π) of order at most 2 for which LIMITED SWITCHINGS (H, π) -COLORING is NP-complete. We first show that two of them are FPT:

Theorem 3.23. *The problem LIMITED SWITCHINGS (H, π) -COLORING is FPT when (H, π) is one of $H_{r,b}^{2b}$ () or $H_{r,-}^{2b}$ ().*

Proof. The signed graph $H_{r,b}^{2b}$ () has the finite duality property by [16], see Lemma 3.5: $G \xrightarrow{p_s} H_{r,b}^{2b}$ if and only if G does not contain a walk $abcd$ where ab and cd are




negative edges and bc is a positive edge. This implies FPT time for LIMITED SWITCHINGS $H_{r,b}^{2b}$ -COLORING by a simple bounded search tree algorithm (Proposition 3.2).

For the graph $H_{r,-}^{2b}$ () , as mentioned in Lemma 3.6, the duality set $\mathcal{F}(H_{r,-}^{2b})$ discovered in [16] is composed of walks of the form $RB^{2p-1}R$ (where R denotes a negative edge, B denotes a positive edge and $p \geq 1$ is an integer) and of closed walks with an odd number of positive edges (*i.e.* cycles in BC_{odd} or UC_{even}). As seen before, if the graph G has such a closed walk then switching will not remove it, thus we can reject.

If the graph has a $RB^{2p-1}R$ walk and is a positive instance, then we claim that we need to switch one of the four vertices incident with the negative edges. Indeed, if we switch only at the vertices inside the positive walk (*i.e.* the vertices not incident with one of the negative edges) then the parity of the number of positive edges will not change and we will still have some maximal odd positive sub-walk, the two edges next to the extremities being negative. Thus we would still have a $RB^{2q-1}R$ walk for some $q \geq 1$.

Hence, since we need to switch at one of these four vertices, we branch on this configuration using the classic bounded search tree technique. This is an FPT algorithm. \square

3.3.3 Limited Switchings (H, π) -Coloring: $W[1]$ -hard cases

The remaining cases, $H_{r,b}^{2rb}$ () , $H_{r,-}^{2rb}$ () , and $H_{r,r}^{2rb}$ () , yield $W[1]$ -hard LIMITED SWITCHINGS (H, π) -COLORING problems, even for input graphs of large girth.

Theorem 3.24. *Let $(H, \pi) \in \{H_{r,b}^{2rb}, H_{r,-}^{2rb}, H_{r,r}^{2rb}\}$. For any integer $g \geq 3$, the problem LIMITED SWITCHINGS (H, π) -COLORING is $W[1]$ -hard, even for signed graphs (G', σ') with girth at least g and which verify $(G', \sigma') \rightarrow_s (H, \pi)$. Under the same conditions, LIMITED SWITCHINGS (H, π) -COLORING cannot be solved in time $f(k)|G|^{o(k)}$ for any computable function f , assuming the ETH.*

We will prove Theorem 3.24 by three reductions from MULTICOLORED INDEPENDENT SET, which is $W[1]$ -complete [162]:

MULTICOLORED INDEPENDENT SET

Parameter: k .

Input: A graph G , an integer k and a partition of $V(G)$ into k sets V_1, \dots, V_k .

Question: Is there a set S of exactly k vertices of G , such that each V_i contains exactly one element of S , that forms an independent set of G ?

Our three reductions (one for each possible choice of x) follow the same pattern. In Section 3.3.3.1, we describe this idea, together with the required properties of the gadgets. In Sections 3.3.3.2, 3.3.3.3 and 3.3.3.4, we show how to construct the gadgets. Since the reduction preserves the parameter and is actually polynomial, the ETH-based lower bound follows from [43].

3.3.3.1 Generic reduction

Let (G, k) be an instance of MULTICOLORED INDEPENDENT SET, and denote by V_1, \dots, V_k the partition of G . Let us construct a signed graph (G', σ) . We begin by creating for each V_i a *partition gadget* (G_i, σ_i) in (G', σ) . This gadget has $|V_i|$ special vertices, denoted x_j for $x_j \in V_i$, in order to associate a vertex of G_i with each vertex of V_i . Moreover, (G_i, σ_i) must satisfy the following.

- (P1) We do not have $(G_i, \sigma_i) \xrightarrow{s^p} (H, \pi)$.
- (P2) If we switch exactly one vertex v of (G_i, σ_i) , then the obtained signed graph admits a sign-preserving homomorphism (H, π) if and only if v is one of the special vertices of G_i .
- (P3) G_i has girth at least g .
- (P4) G_i has two *reset vertices* x and y that are different from the x_i 's and such that the signed graph (G_i, σ'_i) obtained from (G_i, σ_i) by switching x and y admits a sign-preserving homomorphism to (H, π) .

Let uv be an edge of G . Recall that u and v can be seen as vertices of the signed graph (G', σ) . We add an *edge gadget* (G_{uv}, σ_{uv}) (containing two vertices u and v) in (G', σ) by identifying the vertex u (resp. v) of (G', σ) with the vertex u (resp. v) of (G_{uv}, σ_{uv}) . This gadget must satisfy the following.

- (E1) Let (G_{uv}, σ'_{uv}) be the graph obtained from (G_{uv}, σ_{uv}) by switching a subset S of $\{u, v\}$. If $S \neq \{u, v\}$, then $(G_{uv}, \sigma'_{uv}) \not\xrightarrow{s^p} (H, \pi)$.
- (E2) Assume that $u \in V_i$ and $v \in V_j$ and let (P, ρ) be the signed graph obtained from $(G_{uv}, \sigma_{uv}) \cup (G_i, \sigma_i) \cup (G_j, \sigma_j)$ by switching u and v . Then, we do *not* have $(P, \rho) \xrightarrow{s^p} (H, \pi)$.
- (E3) G_{uv} has girth at least g .
- (E4) In G_{uv} , u and v are at distance at least g .

This ends the construction of (G', σ) . Note in particular that every vertex of G is present in G' .

We say that a set S of vertices of G is *valid* if, when seen as a subset of $V(G')$, it contains at most one special vertex in each edge gadget. We need one last condition about (G', σ) .

- (SP) If, after switching a valid set in (G', σ) , the obtained graph does not map to (H, π) , then this is because a partition gadget or an edge gadget does not map to (H, π) (that is, each minimal obstruction is entirely contained in an edge gadget or a partition gadget).

We can now prove that our reduction is valid.

Proposition 3.25. *$((G', \sigma), k)$ is a positive instance of LIMITED SWITCHINGS (H, π) -COLORING if and only if (G, k) is a positive instance of MULTICOLORED INDEPENDENT SET.*

Proof. Assume we can switch at most k vertices of (G', σ) such that the obtained signed graph admits a sign-preserving homomorphism to (H, π) . Let S be the set of those vertices. We claim that S is a valid set of (G', σ) . First note that, due to (P1), S must contain at least one vertex in each V_i . This enforces $|S| = k$, thus S contains exactly one vertex v_i in each V_i . By (P2), each of these v_i 's has to be one of the special vertices of G_i . This means that S contains only vertices that are present in G .

We claim that S induces an independent set in G . Assume by contradiction that there is an edge uv in G with $u, v \in S$. Then, by construction, there is an edge gadget whose special vertices are u and v , such that the edge gadget and the two partition gadgets

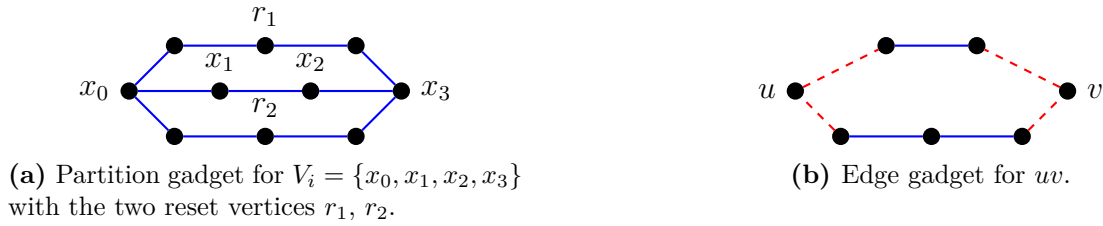



Figure 3.7: Partition and edge gadgets in the $H_{r,b}^{2b}$ -reduction when $g = 3$.

associated with u and v map to (H, π) when we switch only u and v , contradicting $(E2)$. (Note that S does not contain any other vertex of the edge gadget nor any other vertex of the partition gadgets.) Therefore, G has an independent set of size k containing exactly one vertex in each set V_i .

Conversely, assume that G has an independent set S intersecting each V_i at one vertex. Then, we denote by (G', σ') the signed graph obtained by switching every vertex of S in (G, σ) . By construction, this is a valid set, hence by (SP) every obstruction for mapping to (H, π) in (G', σ') is actually contained in some gadget. However, it cannot be contained in a partition gadget due to $(P2)$, nor in an edge gadget due to $(E1)$. Therefore, we have $(G', \sigma') \xrightarrow{p_s} (H, \pi)$. \square

Observe moreover that, due to $(P3)$, $(E3)$ and $(E4)$, G' has girth at least g . Moreover, let S be the set containing all reset vertices of (G', σ) . Let (G', σ') be the signed graph obtained by switching every vertex of S . By $(P4)$, no partition gadget in H contains an obstruction. Furthermore, no edge gadget contains an obstruction by $(E1)$. Therefore, using (SP) , we obtain that H does not contain any obstruction, hence $(G', \sigma') \xrightarrow{p_s} (H, \pi)$. Thus to prove Theorem 3.24 it suffices to construct the gadgets.

3.3.3.2 Gadgets for $H_{r,r}^{2rb}$

We now describe the gadgets for LIMITED SWITCHINGS $H_{r,r}^{2rb}$ -COLORING (). As mentioned in Lemma 3.7, for every signed graph (G, σ) , we have $(G, \sigma) \xrightarrow{p_s} H_{r,r}^{2rb}$ if and only if (G, σ) does not contain an all-positive odd cycle.

The partition gadget (G_i, σ_i) is an all-positive cycle of length $2g$ if g and $|V_i|$ have the same parity (resp. $2g + 2$ is they do not have the same parity) with a positive chord of order $|V_i|$ between two antipodal vertices. The special vertices are those on the chord (see Figure 3.7(a)). The reset vertices are defined as any two vertices on the initial cycle (excluding the two vertices connected to the chord), one on each side of the chord.

Property $(P3)$ directly follows from the construction. Moreover, since G_i contains an all-positive odd cycle, we have $(P1)$. If we switch exactly one vertex in G_i , then either this vertex is a special vertex and the obtained graph does not have any all-positive odd cycle (and thus maps to $H_{r,r}^{2rb}$), or it is not a special vertex and there is still an all-positive odd cycle. Therefore, property $(P2)$ also holds.

Finally, if we switch the two reset vertices, then there is no longer any all-positive odd cycle, thus $(P4)$ also holds.

We now consider the edge gadget. It is formed by an all-positive odd cycle of length $2g + 1$ where two vertices u, v at distance g have been switched (see Figure 3.7(b)). These vertices are the special vertices of the gadget. By construction, properties $(E3)$ and $(E4)$ hold. Moreover, consider a set $S \subset \{u, v\}$. The only way to create a graph containing




Figure 3.8: The edge gadget for uv in the $H_{r,-}^{2rb}$ -reduction when $g = 6$.

an all-positive odd cycle by switching the vertices of S is to switch both u and v . This proves (E1). If we switch both special vertices then we do not have $(G_{uv}, \sigma_{uv}) \xrightarrow{p}_s H_{r,r}^{2rb}$, which implies (E2).

It remains to prove Property (SP). Let S be a valid set, and let (P, ρ) be the graph obtained from (G', σ) when switching all vertices of S . Assume that (P, ρ) contains an all-positive odd cycle. Since S is valid set, at most one vertex has been switched in each edge gadget. Therefore, no all-positive odd cycle of (P, ρ) can contain an edge from an edge gadget. It is thus contained in some partition gadget, ensuring that (SP) holds.

3.3.3.3 Gadgets for $H_{r,-}^{2rb}$

We now describe the gadgets for LIMITED SWITCHINGS $H_{r,-}^{2rb}$ -COLORING (). As mentioned in Lemma 3.8, for every signed graph (G, σ) , we have $(G, \sigma) \xrightarrow{p}_s H_{r,-}^{2rb}$ if and only if (G, σ) does not contain a *bad* walk, *i.e.* an odd figure eight $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$ such that all edges $v_{2i}v_{2i+1}$ are positive [16].

The partition gadget (G_i, σ_i) is the same as in the previous case (see Figure 3.7(a)).

The edge gadget is an odd path of length at least g , whose edges are all-positive except for the first two and last two ones (see Figure 3.8).

Since the partition gadget (G_i, σ_i) is the same as for $H_{r,-}^{2rb}$, Property (P3) still holds. Moreover, since all-positive odd cycles still are obstructions, we have (P1).

Observe that if a signed graph (P, ρ) contains an obstruction, then so does its subgraph obtained by removing recursively its leaves. Note that switching exactly one vertex v in (G_i, σ_i) makes its incident edges all-negative. Therefore, v cannot be contained in a bad walk anymore. In this case, the obstruction is contained in a possibly empty signed cycle (C_v, π_v) (obtained by removing from (G_i, σ_i) the vertex v and the leaves of G_i recursively).

If we switch exactly one vertex in (G_i, σ_i) , then either this vertex is a special vertex and (C_v, π_v) is empty or an all-positive even cycle (and thus maps to $H_{r,-}^{2rb}$), or it is not a special vertex and (C_v, π_v) is still an all-positive odd cycle. Therefore, property (P2) also holds.

Finally, if we switch the two reset vertices u and v , then $G_i \setminus \{u, v\}$ is a tree, thus G_i does not contain any obstruction, hence (P4) also holds.

By construction, properties (E3) and (E4) hold. Moreover, observe that the edge gadget does not contain a bad walk since it is a path. Thus (E1) holds. If (P, ρ) is the graph defined in property (E2) then there is a bad walk starting from u , then turning around one odd cycle in the partition gadget containing u , crossing the edge gadget to v , taking a similar turn around an odd cycle of the partition gadget containing v and then going back to u by the edge gadget. So (E2) holds.

It remains to prove (SP). Let S be a valid set, and (G', σ') be the graph obtained from (G', σ') by switching S . Observe that no bad walk contains two consecutive negative edges. Moreover, in (G', σ') , every edge gadget contains two such edges (since its two endpoints cannot be both in S). Therefore, no bad walk crosses an edge gadget (G_{uv}, σ_{uv}) , which implies that no bad walk contains edges in (G_{uv}, σ_{uv}) . Hence, every bad walk is contained in some partition gadget, thus ensuring that (SP) holds.

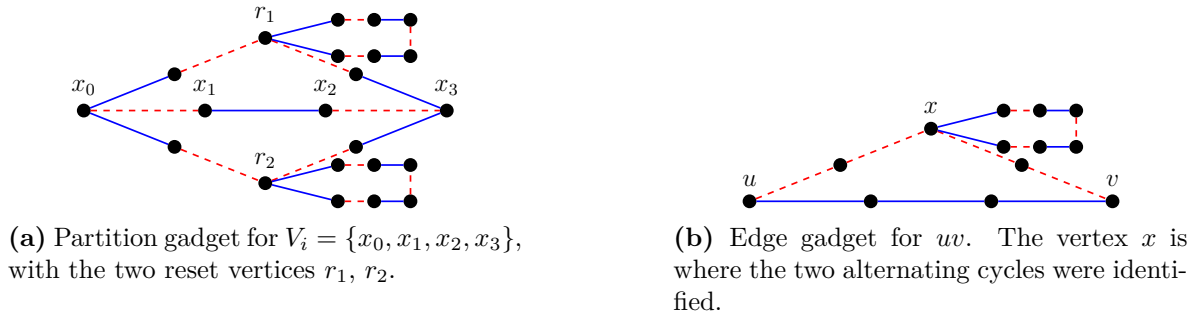


Figure 3.9: Partition and edge gadgets in the $H_{r,b}^{2rb}$ -reduction when $g = 3$.

3.3.3.4 Gadgets for $H_{r,b}^{2rb}$

We now describe the gadgets for LIMITED SWITCHINGS $H_{r,b}^{2rb}$ -COLORING (). As mentioned in Lemma 3.9, for every signed graph (G, σ) , we have $(G, \sigma) \xrightarrow[p]{s} H_{r,b}^{2rb}$ if and only if (G, σ) does not contain an *alternating odd figure eight*, that is, an alternating closed walk $v_0, v_1, \dots, v_{2j}, v_0, v_{2j+2}, \dots, v_{2p-1}, v_0$ for some integers j and p [16].

The partition gadget (G_i, σ_i) is defined by gluing two obstructions with large girth along a path of length $|V_i|$ (see Figure 3.9(a)). More precisely, consider an alternating odd cycle (C, π_C) of size $|V_i| + g$ (or $|V_i| + g + 1$). Note that (C, π_C) contains a vertex u adjacent to two negative edges. We attach an alternating odd cycle (C', π'_C) of length g (or $g + 1$) to u , such that the edges of (C', π'_C) adjacent to u are positive. To obtain (G_i, σ_i) , we take two copies of this obstruction, and glue their respective largest cycle along a path of length $|V_i|$. The vertices of this path are the special vertices of (G_i, σ_i) , and the two copies of u are the reset vertices of (G_i, σ_i) .

The edge gadget is formed by identifying the vertices having their two incident edges of the same sign in two alternating odd cycles of length $2g + 1$, in such a way that the common vertex has two positive edges in one cycle and two negative edges in the other one. To obtain the edge gadget, we switch two vertices u and v , at distance g from each other, in the same cycle of this signed graph (see Figure 3.9(b)).

Observe that (G_i, σ_i) has girth at least g , hence Property (P3) holds. Moreover, by construction, (G_i, σ_i) contains an obstruction, hence (P1) holds. Note that there are exactly two (minimal) obstructions in (G_i, σ_i) , the ones used to construct it. Therefore, if we switch a non-special vertex in (G_i, σ_i) , one of the these obstructions is unchanged, and the obtained graph does not map to $H_{r,b}^{2rb}$. Conversely, assume that we switch a special vertex u of (G_i, σ_i) and there remains an obstruction. Note that all the paths of length two starting from u are now all-positive or all-negative, hence no alternating odd figure eight can go through u . This implies that every alternating odd figure eight in this graph does not use the internal vertices of the glued path. When removing these vertices from (G_i, σ_i) , the former endpoints of the glued path have their incident edges of the same sign, hence they cannot be contained in an alternating odd figure eight. Removing the whole glued path and (recursively) the leaves of G_i gives two disjoint alternating odd cycles, which do not contain any alternating odd figure eight. Thus we have (P2).

Finally, if we switch the two reset vertices of (G_i, σ_i) , all the paths of length 2 starting at these vertices are all-positive or all-negative, hence no alternating odd figure eight goes through them. Removing the reset vertices, and recursively the obtained leaves gives the empty graph. Therefore, there is no alternating odd figure eight in the signed graph

obtained from (G_i, σ_i) by removing the reset vertices, it thus maps to $H_{r,b}^{2rb}$, and (P4) holds.

The construction of the edge gadget ensures that (E3) and (E4) are satisfied. Moreover, if we switch u and v , we obtain an obstruction, ensuring that (E2) holds. Finally, let (G_{uv}, σ'_{uv}) be the graph obtained from (G_{uv}, σ_{uv}) by possibly switching v . Then every path of length two starting at u is all-positive or all-negative, hence no alternating odd figure eight in H contains u . Removing u and leaves of (G_{uv}, σ'_{uv}) yields an alternating odd cycle, which does not contain any alternating odd figure eight. Therefore, (G_{uv}, σ'_{uv}) maps to $H_{r,b}^{2rb}$, and by exchanging u with v , we obtain (E1).

It remains to prove (SP). Let S be a valid set and (G', σ'') be the graph obtained from (G', σ') by switching every vertex of S . Consider an alternating odd figure eight containing an edge from an edge gadget and an edge from a partition gadget. This walk goes through a vertex $u \in V_i$ such that the edge before u in the walk lies in (G_i, σ_i) and the other one lies in some (G_{uv}, σ_{uv}) . If $u \in S$, the paths of length 2 starting from u in (G_i, σ_i) are all-positive or all-negative. Conversely, if $u \notin S$, the paths of length 2 starting at u in (G_{uv}, σ_{uv}) are all-positive or all-negative. In both cases we reach a contradiction with the existence of a bad walk going through u . Therefore, every alternating odd figure eight of (G', σ'') is contained either in an edge gadget or in a partition gadget.

3.4 Vertex Deletion Signed- (H, π) -Coloring

In the previous section, we considered “switching” as the modification operation on signed graphs. As seen in Chapter 2, we can color signed graphs using sign-preserving homomorphisms or using homomorphisms of signed graphs. Hence for a given signed graph (H, π) , we have two coloring problems: (H, π) -COLORING and SIGNED- (H, π) -COLORING.

Until now, we applied modifications operations to sign-preserving homomorphisms (or homomorphisms of 2-edge-colored graphs) as the homomorphism type on signed graphs (or equivalently, 2-edge-colored graphs). From now on, we will use homomorphisms of signed graphs (*i.e.* homomorphisms with an unlimited number of switchings) and consider modification problems of SIGNED- (H, π) -COLORING.

Of the three previous modification types, vertex deletion, edge deletion and switching, only the first two are interesting in this setting since homomorphisms of signed graphs are built in with an unlimited number of switchings. In this section, we focus on vertex deletion for the problem SIGNED- (H, π) -COLORING.

VERTEX DELETION SIGNED- (H, π) -COLORING

Parameter: k

Input: A signed graph (G, σ) and an integer k .

Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S \xrightarrow{s} (H, \pi)$?

As always, if (H, π) is not some type of core then the problem reduces to another problem with a smaller target graph. Here the relevant notion of core is the notion of s-core. By Theorem 2.22, if (H, π) is an s-core with at least three edges, then VERTEX DELETION SIGNED- (H, π) -COLORING is NP-complete even for $k = 0$.


Up to equivalence, there are seven s-cores which have at most two edges: H_-^1 (●), H_{rb}^1 (●—●), H_b^1 (●—●), H_r^1 (●—●), $H_{-, -}^{2b}$ (●—●), $H_{-, -}^{2rb}$ (●—●) and $H_{r,b}^{2-}$ (●—●). Note that $H_{-, -}^{2b}$ (●—●) and $H_{-, -}^{2r}$ (●—●) are equivalent: switching one of the two vertices transform one signed graph into the other.

The following theorem completely characterizes the complexity of VERTEX DELETION SIGNED- (H, π) -COLORING both in terms of P/NP-complete dichotomy and in terms of parameterized complexity.

Theorem 3.26. *Let (H, π) be an s -core. The following statements hold.*



1. *The problem VERTEX DELETION SIGNED- H_{rb}^1 -COLORING is in P.*
2. *The problem VERTEX DELETION SIGNED- (H, π) -COLORING is NP-complete when (H, π) is one of H_-^1 , H_b^1 , H_r^1 , $H_{-,-}^{2b}$, $H_{-,-}^{2rb}$ or $H_{r,b}^{2-}$.*
3. *The problem VERTEX DELETION SIGNED- (H, π) -COLORING is NP-complete even for $k = 0$ when $(H, \pi) \notin \{H_-^1, H_{rb}^1, H_b^1, H_r^1, H_{-,-}^{2b}, H_{-,-}^{2rb}, H_{r,b}^{2-}\}$.*
4. *The problem VERTEX DELETION SIGNED- (H, π) -COLORING is FPT and can be solved in $2^{O(k)} |V(G)|^{O(1)}$ time when (H, π) is one of H_-^1 , H_b^1 , H_r^1 or $H_{-,-}^{2rb}$.*
5. *The problem VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING is FPT and can be solved in $2^{O(k^3)} |V(G)|^{O(1)}$ time.*
6. *The problem VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING is FPT and can be solved in $2^{O(k^2)} |V(G)|^{O(1)}$ time.*

Proof.


1. As always, every signed graph admits a homomorphism to the signed graph H_{rb}^1 (). Hence we can accept any instance of VERTEX DELETION SIGNED- H_{rb}^1 -COLORING for which $k \geq 0$.
2. By Theorem 2.17, for every signed graph (H, π) , the problem VERTEX DELETION SIGNED- (H, π) -COLORING is equivalent to the problem VERTEX DELETION DSG (H, π) -COLORING where DSG (H, π) is the double switching graph of (H, π) seen as a 2-edge-colored graph (see Theorem 2.17).

Hence, by Theorem 3.12, if $(H, \pi) \in \{H_-^1, H_b^1, H_r^1, H_{-,-}^{2b}, H_{-,-}^{2rb}, H_{r,b}^{2-}\}$ then VERTEX DELETION SIGNED- (H, π) -COLORING is NP-complete.

3. This follows directly from Theorem 2.22.
4. The problem VERTEX DELETION SIGNED- H_-^1 -COLORING is equivalent to VERTEX COVER, and is thus FPT.


For the signed graph H_b^1 () , note that DSG (H_b^1) is a signed graph isomorphic to the 2-edge-colored graph $H_{b,b}^{2r}$ (). By Theorem 2.17 and Theorem 3.20, VERTEX DELETION SIGNED- H_b^1 -COLORING is FPT and can be solved in $2^{O(k)} |V(G)|^{O(1)}$ time.

The problem VERTEX DELETION SIGNED- H_r^1 -COLORING is also FPT as it is equivalent to VERTEX DELETION SIGNED- H_b^1 -COLORING on input $(G, -\sigma)$ and k by Observation 2.14.

For the signed graph $H_{-,-}^{2rb}$ () , the sign of the edges do not matter in the homomorphism: $(G, \sigma) \rightarrow_s H_{-,-}^{2rb}$ if and only if G is bipartite. Hence VERTEX DELETION SIGNED- $H_{-,-}^{2rb}$ -COLORING is equivalent to ODD CYCLE TRANSVERSAL which is FPT and can be solved in $2^{O(k)} |V(G)|^{O(1)}$ time (see [166, 127]).

5. The remaining problems are more complicated. The problem VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING, on an input composed of a signed graph (G, σ) and an integer k , consists in finding a set S of at most k vertices of G such that $\chi_s((G, \sigma) - S) \leq 2$, or equivalently such that $(G, \sigma) - S$ is a bipartite balanced signed graph. In some sense, this problem consists in solving VERTEX DELETION SIGNED- H_b^1 -COLORING and VERTEX DELETION SIGNED- $H_{-, -}^{2rb}$ -COLORING at the same time. Unfortunately, $\text{DSG}(H_{-, -}^{2b})$ has four vertices and we cannot apply the techniques of Theorem 3.20 to solve the problem. Indeed each vertex of $H_{-, -}^{2b}$ would need to be represented by two variables and we cannot ensure that a solution of the corresponding VARIABLE DELETION ALMOST 2-SAT instance deletes variables in pairs.

Nonetheless, this problem is very similar to ODD CYCLE TRANSVERSAL except that we need to delete not only odd cycles but also unbalanced cycles. Our algorithm uses the “iterative compression” technique (see [53, Chapter 4]) which was created to solve ODD CYCLE TRANSVERSAL (see [166]). We present this algorithm in Section 3.4.1.

6. The problem VERTEX DELETION SIGNED- $H_{r, b}^{2r}$ -COLORING () on input signed graph (G, σ) and integer k , consists in finding a set S of at most k vertices of G such that each connected components of $(G, \sigma) - S$ is either balanced or antibalanced. In Section 3.4.2, we use iterative compression to construct our FPT algorithm.

Note that, we completely characterize the complexity of all s-cores. □

3.4.1 Vertex Deletion Signed- $H_{-, -}^{2b}$ -Coloring: an FPT algorithm using iterative compression

We want to prove that VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING is FPT.

First note that, other than the empty graph of order 2, $H_{-, -}^{2b}$ is the only simple signed graph of order 2 (up to equivalence). Hence $\chi_s(G, \sigma) \leq 2$ if and only if $(G, \sigma) \rightarrow_s H_{-, -}^{2b}$. Since $\chi(G) \leq \chi_s(G, \sigma)$, $(G, \sigma) \rightarrow H_{-, -}^{2b}$ implies that G is bipartite. Moreover since $H_{-, -}^{2b}$ has no unbalanced closed walks, $(G, \sigma) \rightarrow H_{-, -}^{2b}$ implies that (G, σ) is balanced. Finally, if (G, σ) is bipartite balanced, then (G, σ) is equivalent to (G, \emptyset) and since $\chi_s(G, \emptyset) = \chi(G)$, $\chi_s(G, \sigma) \leq 2$. Hence $\chi_s(G, \sigma) \leq 2$ if and only if $(G, \sigma) \rightarrow_s H_{-, -}^{2b}$ if and only if (G, σ) is bipartite balanced. This leads us to reformulate the problem as follows.


VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING **Parameter:** k
Input: A signed graph (G, σ) and an integer k .
Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S$ is a bipartite balanced signed graph?

To this end, we define the following two problems.


DISJOINT VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING **Parameter:** k
Input: A signed graph (G, σ) , an integer k and a set S' of at most $k + 1$ vertices of G such that $(G, \sigma) - S'$ is a bipartite balanced signed graph.
Question: Is there a set S of at most k vertices of G such that $S \cap S' = \emptyset$ and $(G, \sigma) - S$ is a bipartite balanced signed graph?

ANNOTATED BIPARTITE BALANCED COLORING

Parameter: k
Input: A bipartite balanced signed graph (G, σ) , an integer k and four sets of vertices B_1, B_2, B_+, B_- .

Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S$ admits a sign-preserving homomorphism φ to $H_{b,b}^{2r}$ () (where the two vertices of $H_{b,b}^{2r}$ are named \oplus and \ominus) and a proper 2-coloring $c : V(G) \setminus S \rightarrow \{1, 2\}$ verifying the following properties:

1. $\varphi(B_+ \setminus S) = \{\oplus\}$, $\varphi(B_- \setminus S) = \{\ominus\}$,
2. $c(B_1 \setminus S) = \{1\}$ and $c(B_2 \setminus S) = \{2\}$?

Note that $(G, \sigma) \xrightarrow{p_s} H_{b,b}^{2r}$ if and only if (G, σ) is balanced as $H_{b,b}^{2r}$ is the double switching graph of H_b^1 (), thus justifying the name of the problem.

Our goal is to use DISJOINT VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING in order to solve VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING and to use ANNOTATED BIPARTITE BALANCED COLORING in order to solve DISJOINT VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING. In order to construct an FPT algorithm for ANNOTATED BIPARTITE BALANCED COLORING, we consider the following problem.

VERTEX MULTICUT

Parameter: k
Input: A graph G , ℓ ordered pairs $(s_1, t_1), \dots$ and (s_ℓ, t_ℓ) of vertices of G and an integer k .

Question: Is there a set S of at most k vertices of G such that, for every $i \in \llbracket 1, \ell \rrbracket$, if $s_i, t_i \notin S$, then the graph $G - S$ does not contain a path from s_i to t_i ?

As $\ell = O(|V(G)|^2)$, the following result follows from [151].

Theorem 3.27 ([151]). *The problem VERTEX MULTICUT can be solved in $2^{O(k^3)} |V(G)|^{O(1)}$ time.*

Our first goal is to prove the following lemma.

Lemma 3.28. *The following statements hold.*


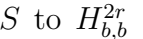
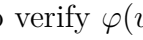
1. *The problem ANNOTATED BIPARTITE BALANCED COLORING can be solved in $2^{O(k^3)} |V(G)|^{O(1)}$ time.*
2. *The problem DISJOINT VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING can be solved in $2^{O(k^3)} |V(G)|^{O(1)}$ time.*

Proof of Lemma 3.28.1. Consider an instance of ANNOTATED BIPARTITE BALANCED COLORING composed of a bipartite balanced signed graph (G, σ) , an integer k and four subsets of $V(G)$: B_1, B_2, B_+ and B_- .

Since G is bipartite, G admits a 2-coloring $c : V(G) \rightarrow \{1, 2\}$. We want to construct a 2-coloring c' for which non-deleted vertices of B_1 have color 1 and non-deleted vertices of B_2 have color 2. Suppose that there exists a solution S of at most k vertices of G , and let $v \in V(G - S)$. If $v \in B_2$ and $c(v) = 1$ then we need to change the color of v from c to c' . If we change the color of v , we also need to change the color of its neighbors in $G - S$ to keep the property of the coloring. In particular, the vertices of $\text{Change}_{bip} = (B_1 \cap c^{-1}(\{2\})) \cup (B_2 \cap c^{-1}(\{1\}))$ must be removed or must have distinct colors

by c and c' . To the contrary, the vertices of $\text{Stay}_{bip} = (B_1 \cap c^{-1}(\{1\})) \cup (B_2 \cap c^{-1}(\{2\}))$ must be removed or have the same color in c and c' . Hence if S is a solution, then for every two vertices $u \in \text{Change}_{bip} \setminus S$ and $v \in \text{Stay}_{bip} \setminus S$, there must be no path between u and v in $G - S$.

Moreover, if S is a set of at most k vertices of G such that for any two vertices $u \in \text{Change}_{bip} \setminus S$ and $v \in \text{Stay}_{bip} \setminus S$, there is no path between u and v in $G - S$, then there exists a 2-coloring c' of $G - S$ for which $c'(B_1 \setminus S) = \{1\}$ and $c'(B_2 \setminus S) = \{2\}$. To construct c' , it suffices to change the color (with respect to the 2-coloring c) of every vertex of a connected component of $G - S$ containing a vertex of Change_{bip} .

As (G, σ) is balanced, there exists a sign-preserving homomorphism φ from (G, σ) to $H_{b,b}^{2r}$ () . We want to find another sign-preserving homomorphism φ' from $(G, \sigma) - S$ to $H_{b,b}^{2r}$ () where S is the desired solution. As for the bipartite case, some vertices of G must change to have distinct images by φ and φ' (or must be deleted) and other must have the same image by φ and φ' (or must be deleted). Suppose that $\varphi(v) \neq \varphi'(v)$, for some vertex v , then each non-deleted neighbor u of v must also verify $\varphi(u) \neq \varphi'(u)$ so that the edge uv maps to an edge of $H_{b,b}^{2r}$ () with sign $\sigma(uv)$. Hence we can apply the same arguments as for the bipartite condition. We can define two sets $\text{Change}_{sw} = (B_+ \cap \varphi^{-1}(\{\ominus\})) \cup (B_- \cap \varphi^{-1}(\{\oplus\}))$ and $\text{Stay}_{sw} = (B_+ \cap \varphi^{-1}(\{\oplus\})) \cup (B_- \cap \varphi^{-1}(\{\ominus\}))$. A solution S must disconnects vertices of $\text{Change}_{sw} \setminus S$ from vertices of $\text{Stay}_{sw} \setminus S$. Moreover, any set S of at most k vertices of G which disconnect vertices of $\text{Change}_{sw} \setminus S$ from vertices of $\text{Stay}_{sw} \setminus S$ implies the existence of φ' as described in the problem definition. To construct φ' it suffices to change the image (with respect to the φ) of every vertex of a connected component of $G - S$ containing a vertex of Change_{sw} .

To conclude, in order to solve ANNOTATED BIPARTITE BALANCED COLORING, it suffices to find a set S which disconnects vertices of $\text{Change}_{bip} \setminus S$ from vertices of $\text{Stay}_{bip} \setminus S$ and disconnects vertices of $\text{Change}_{sw} \setminus S$ from vertices of $\text{Stay}_{sw} \setminus S$.

We construct the set T of ordered pairs of vertices of G as follows, T is composed of all pairs (s, t) where $s \in \text{Change}_{bip}$ and $t \in \text{Stay}_{bip}$ and all pairs (s, t) where $s \in \text{Change}_{sw}$ and $t \in \text{Stay}_{sw}$. The graph G along with the set of ordered pairs of T and the integer k forms an instance of VERTEX MULTICUT. A subset S of $V(G)$ is a solution of this instance of VERTEX MULTICUT if and only if it is a solution of our instance of ANNOTATED BIPARTITE BALANCED COLORING. Hence, we can solve ANNOTATED BIPARTITE BALANCED COLORING in $2^{O(k^3)} |V(G)|^{O(1)}$ time by Theorem 3.27. \square

Proof of Lemma 3.28.2. Consider an instance of DISJOINT VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING composed of a signed graph (G, σ) , an integer k and a set S' of at most $k + 1$ vertices of G such that $(G, \sigma) - S'$ is a bipartite balanced signed graph. Let $(G[S'], \sigma[S'])$ be the signed subgraph of (G, σ) induced by S' . If $(G[S'], \sigma[S'])$ is not bipartite balanced then the problem does not have a solution and we can reject this instance.

Suppose that S is a solution for our instance. The signed graph $(G, \sigma) - S$ is balanced bipartite. Hence, $(G, \sigma) - S$ has a 2-coloring (resp. a sign-preserving homomorphism to $H_{b,b}^{2r}$). This coloring (resp. sign-preserving homomorphism) has a restriction c^* (resp. φ^*) to $(G[S'], \sigma[S'])$.

Given a 2-coloring c of $(G[S'], \sigma[S'])$ and a sign-preserving homomorphism φ from $(G[S'], \sigma[S'])$ to $H_{b,b}^{2r}$, we want to be able to test whether we can remove some set of vertices $S \subseteq V(G) \setminus S'$ in such a way that c and φ extend to the vertices of $V(G) \setminus (S \cup S')$.

Let (G', σ') be the signed subgraph of (G, σ) induced by $V(G) \setminus S'$. We define the following sets:

$$\begin{aligned} B_1(c) &= \{u \in V(G') : \exists v \in S' \cap N(u), c(v) = 2\}, \\ B_2(c) &= \{u \in V(G') : \exists v \in S' \cap N(u), c(v) = 1\}, \\ B_+(\varphi) &= \{u \in V(G') : \exists v \in S' \cap N(u), \varphi(v) = \sigma(uv)\}, \\ B_-(\varphi) &= \{u \in V(G') : \exists v \in S' \cap N(u), \varphi(v) = -\sigma(uv)\}. \end{aligned}$$

Note that in order to extend c to the rest of G , a vertex in $B_1(c)$ must be removed or must be assigned color 1. The same remarks holds for the other sets. Hence a set S is a solution of our instance of DISJOINT VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING if and only if it is a solution of the instance of ANNOTATED BIPARTITE BALANCED COLORING composed of (G', σ') , the integer k and the four sets $B_1(c)$, $B_2(c)$, $B_+(\varphi)$ and $B_-(\varphi)$ for some choice of c and φ . In particular $c = c^*$ and $\varphi = \varphi^*$ (if they exist) is a good choice for these two functions. Note that the choices of c^* and φ^* were arbitrary and it may be possible that other choices might correspond to a solution of ANNOTATED BIPARTITE BALANCED COLORING.

Therefore, it suffices to test for every pair $\{c, \varphi\}$, where c is a 2-coloring of $(G[S'], \sigma[S'])$ and φ is a sign-preserving homomorphism from $(G[S'], \sigma[S'])$ to $H_{b,b}^{2r}$, whether there is a solution to the corresponding instance of ANNOTATED BIPARTITE BALANCED COLORING. If there is no solution then we can reject as this contradicts the existence of c^* and φ^* . If there is a solution S then we can accept and return this solution.

Note that the number of choices for the pair $\{c, \varphi\}$ is upper bounded by $(2^{k+1})^2$. Hence we can solve DISJOINT VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING in $(2^{k+1})^2 \times 2^{O(k^3)} |V(G)|^{O(1)} = 2^{O(k^3)} |V(G)|^{O(1)}$ time. \square

We can now prove Theorem 3.26.5 using the lemma.

Proof of Theorem 3.26.5. Consider an instance \mathcal{I} of VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING composed of a signed graph (G, σ) and an integer k . We solve the problem by induction on the number of vertices. If G has at most k vertices than we can accept and return $V(G)$. This is our base case.

Let x be a vertex of G . By induction, we can test whether there is a solution of VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING on the instance \mathcal{I}' composed of the signed graph $(G, \sigma) - x$ and the integer k . If there is no solution for \mathcal{I}' then \mathcal{I} does not have a solution and we can reject. Hence we can suppose that there is a solution S'' for the instance \mathcal{I}' . Note that, by definition, $(G, \sigma) - (S'' \cup \{x\})$ is a bipartite balanced signed graph but $S' = S'' \cup \{x\}$ might contain $k + 1$ vertices. In this case, we need to find a smaller solution. If \mathcal{I} has a solution S^* then we try to determine the (possibly empty) set $S^* \cap S'$. For every possible subset X of S' , we test whether this set can correspond to the intersection of S^* and S' .


To do that it suffices to test whether there is a solution to the problem DISJOINT VERTEX DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING on the instance composed of the signed graph $(G, \sigma) - X$, the integer $k - |X|$ and the set $S' \setminus X$. Note that $(G, \sigma) - X - (S' \setminus X)$ is a bipartite balanced signed graph. If such a solution S exists then $S \cup X$ is a solution of \mathcal{I} and we can accept. Otherwise, the set X does not correspond to the intersection of S^* and S' .

If none of the 2^{k+1} possible intersections yields a solution then we can reject. Hence we can solve \mathcal{I} with a solution to \mathcal{I}' in $(2^{k+1}) \times 2^{O(k^3)} |V(G)|^{O(1)}$ time. As we need to perform

$O(|V(G)|)$ recursive calls which correspond to the removal of x , we need to multiply the complexity by $O(|V(G)|)$. In the end, it takes a $2^{O(k^3)} |V(G)|^{O(1)}$ time to solve our problem. \square

3.4.2 Vertex Deletion Signed- $H_{r,b}^{2-}$ -Coloring: an FPT algorithm using iterative compression


The problem VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING can be reformulated in the following way: how many vertices do we need to remove from the input signed graph (G, σ) so that we obtain a signed graph with no unbalanced even closed walk? Note that we cannot replace “closed walk” by “cycle” in the previous characterization. Due to this, we cannot apply the same arguments as in Lemma 3.28.1. Fortunately, we can still use iterative compression in order to reduce our problem.

The algorithm for this section starts in a similar fashion to the algorithm of the previous section. Consider from now on that $H_{r,b}^{2-}$ () has for vertex set $\{\oplus, \ominus\}$ where \oplus is the vertex with the positive loop and \ominus is the vertex with the negative loop. Remark that we can solve VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING with $O(2^k |V(G)|)$ calls to DISJOINT VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING as in the previous section.

DISJOINT VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING **Parameter:** k
Input: A signed graph (G, σ) , an integer k and a set S' of at most $k + 1$ vertices of G such that $(G, \sigma) - S' \rightarrow_s H_{r,b}^{2-}$.
Question: Is there a set S of at most k vertices of G such that $S \cap S' = \emptyset$ and $(G, \sigma) - S \rightarrow_s H_{r,b}^{2-}$?

Let $D(n, k)$ be the maximal time taken by our algorithm to solve an instance of DISJOINT VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING with parameter at most k and where the input signed graph has order at most n .

To solve the disjoint version of our problem let us introduce new problems. The first of these intermediate problems is ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING.

ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING **Parameter:** k
Input: A signed graph (G, σ) , an integer k and two disjoint sets of vertices B_+ , B_- such that $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$.
Question: Is there a subset S of $V(G)$ of size at most k such that $(G, \sigma) - S$ admits a homomorphism φ to $H_{r,b}^{2-}$ () , verifying $S \cap (B_+ \cup B_-) = \emptyset$, $\varphi(B_+) = \oplus$ and $\varphi(B_-) = \ominus$?

This problem can have two types of instances: instances which behave well (called *nice instances*) and the others. Let \mathcal{I} be an instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of a signed graph (G, σ) , an integer k and two disjoint sets B_+ and B_- . The instance \mathcal{I} is *nice* if and only if, for every $u \in V(G) \setminus (B_+ \cup B_-)$, either u is disconnected from the vertices of $B_+ \cup B_-$ or, there exist a (u, B_+) -path and a (u, B_-) -path whose internal vertices belong to $V(G) \setminus (B_+ \cup B_-)$. Unfortunately, in general, we cannot suppose that our instances are nice. Nice instances will appear as a byproduct of some recursive calls of the algorithm. This notion of nice instance is important as not making the distinction gives a non-FPT run-time for our algorithm.


Let $A(n, k)$ be the maximal time taken by our algorithm to solve an instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING with parameter at most k and where the input signed graph has order at most n . Let $A_{nice}(n, k)$ be the maximal time taken by our algorithm to solve a nice instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING with parameter at most k and where the input signed graph has order at most n .

We define two other problems.

CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING

Parameter: k


Input: A signed graph (G, σ) , an integer k and two non-empty disjoint sets of vertices B_+, B_- such that $G - (B_+ \cup B_-)$ is connected and $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$.

Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S$ admits a homomorphism φ to $H_{r,b}^{2-}$ () , verifying $S \cap (B_+ \cup B_-) = \emptyset$, $\varphi(B_+) = \oplus$ and $\varphi(B_-) = \ominus$?

CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING

Parameter: k

Input: A signed graph (G, σ) , an integer k and a non-empty set of vertices B_+ such that $(G, \sigma) - B_+$ is connected and $(G, \sigma) - B_+ \rightarrow_s H_{r,b}^{2-}$.

Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S$ admits a homomorphism φ to $H_{r,b}^{2-}$ () , verifying $S \cap B_+ = \emptyset$ and $\varphi(B_+) = \oplus$?


There are a few things to note. The connectivity requirement for these two problems is not on the input signed graph (G, σ) but on the graph $G - (B_+ \cup B_-)$. Intuitively, the vertices of $B_+ \cup B_-$ can be interpreted as “hints” for the homomorphism which cannot be interacted with: we cannot remove them and they do not intervene in the connectivity. Of the two, the problem CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING behaves the nicest. In particular, nice instances of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING will be reduced to a number of instances of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. Unfortunately, the usual instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING will also reduce to some instances CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. This problem has less constraints than CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING which makes its resolution more complicated. In particular, it is possible that our algorithm for CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING reduces to ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING with an instance of roughly the same size. Fortunately, in this case, the instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING is nice, which avoids infinite loops and ensure our FPT complexity.

Finally, even though CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING is defined for $B_- = \emptyset$ and $B_+ \neq \emptyset$, we could define the following symmetrical problem.

CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING (v2)

Parameter: k

Input: A signed graph (G, σ) , an integer k and a non-empty set of vertices B_- such that $(G, \sigma) - B_-$ is connected and $(G, \sigma) - B_- \rightarrow_s H_{r,b}^{2-}$.

Question: Is there a set S of at most k vertices of G such that $(G, \sigma) - S$ admits a homomorphism φ to $H_{r,b}^{2-}$ () , verifying $S \cap B_- = \emptyset$ and $\varphi(B_-) = \ominus$?

In practice, we consider that the problem CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING (v2) is the same as CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. This means that we do not make the distinction between the two in the rest

of this section.

Let $\text{CA}(n, k)$ be the maximal time taken by our algorithm to solve an instance of $\text{CONNECTED ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ with parameter at most k and where the input signed graph has order at most n . Let $\text{CHA}(n, k)$ be the maximal time taken by our algorithm to solve an instance of $\text{CONNECTED HALF ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ with parameter at most k and where the input signed graph has order at most n .

The rest of this section is structured as follows. We present our algorithm for $\text{DISJOINT VERTEX DELETION SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ and $\text{ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ in Section 3.4.3 and Section 3.4.4 respectively. Section 3.4.5 contains a number of definitions used for solving the last two problems. Section 3.4.6 presents how we handle the problem $\text{CONNECTED HALF ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$. Finally, Section 3.4.7 contains the algorithm for $\text{CONNECTED ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$, this section is where most of the work of the algorithm is done. In Section 3.4.8, we compute the complexity of our algorithm.

Finally, note that when computing the complexity, we use the following abuse of notation $f(n, k) \leq O(g(n, k)) + h(n, k)$ to signify that there exists a function $e(n, k) = O(g(n, k)) + h(n, k)$ such that $f(n, k) \leq e(n, k)$ for all $n \in \mathbb{N}$ and $k \in \mathbb{N}$.

3.4.3 Solving Disjoint Vertex Deletion Signed- $H_{r,b}^{2-}$ -Coloring

Let \mathcal{I} be an instance of $\text{DISJOINT VERTEX DELETION SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ composed of a signed graph (G, σ) , an integer k and a set S' of at most $k + 1$ vertices such that $(G, \sigma) - S' \rightarrow_s H_{r,b}^{2-}$.

For every homomorphism $\varphi : (G, \sigma)[S'] \rightarrow_s H_{r,b}^{2-}$, we create the instance $\mathcal{I}'(\varphi)$ of $\text{ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ composed of (G, σ) , k and the two disjoint sets $B_+ = \varphi^{-1}(\oplus)$ and $B_- = \varphi^{-1}(\ominus)$.

Suppose that S is a solution to \mathcal{I} and let $\varphi_S : (G, \sigma) - S \rightarrow_s H_{r,b}^{2-}$. In particular, φ_S has a restriction φ'_S to S' . Note that S is a solution to $\mathcal{I}'(\varphi'_S)$ by definition of φ'_S . Moreover, if S'' is a solution to $\mathcal{I}'(\varphi)$ then S'' is also a solution of \mathcal{I} .

For every homomorphism $\varphi : (G, \sigma)[S'] \rightarrow_s H_{r,b}^{2-}$, our algorithm test whether $\mathcal{I}'(\varphi)$ is a positive instance or not. If there is a positive instance then our algorithm accepts, otherwise it rejects. There are at most 2^{k+1} homomorphisms φ from $(G, \sigma)[S']$ to $H_{r,b}^{2-}$, and we can test in $O(k^2)$ time, for each function from S' to $V(H_{r,b}^{2-})$, whether it is a homomorphism or not.

Lemma 3.29. *We have*

$$D(n, k) \leq 2^{k+1}(O(k^2) + A(n, k)).$$

3.4.4 Solving Annotated Signed- $H_{r,b}^{2-}$ -Coloring

Let \mathcal{I} be an instance of $\text{ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$ composed of a signed graph (G, σ) , an integer k and two disjoint sets B_+ and B_- such that $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$. See Figure 3.10 for a schematic view of some instance of $\text{ANNOTATED SIGNED-}H_{r,b}^{2-}\text{-COLORING}$. The instance presented in Figure 3.10 is not nice due to the presence of (G_4, σ_4) which is not connected to B_- .

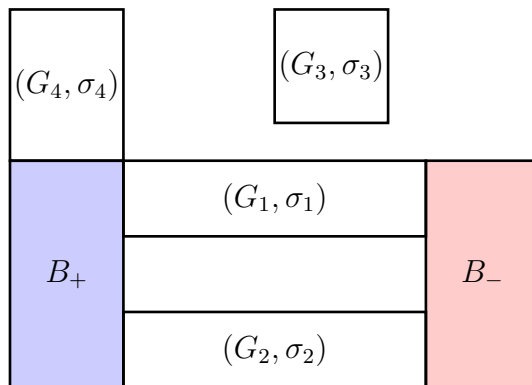


Figure 3.10: A possible instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. Each rectangle (with a label inside) represents a connected signed graph. Two rectangles sharing a border are connected. There are no edges between signed graphs whose rectangles do not intersect.

If $k < 0$, then we can reject. If $k = 0$, then we can solve our problem in polynomial time, it suffices to check whether each connected component maps to \oplus or \ominus . Our algorithm separates our problem into a number of sub-problems where $G - (B_+ \cup B_-)$ is connected.

Let \mathcal{CC} be the set of connected components of $G - (B_+ \cup B_-)$. For $C \in \mathcal{CC}$, and $k' \leq k$, we construct the instance $\mathcal{I}'(C, k')$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of the signed graph $(G, \sigma)[V(C) \cup N_G(V(C))]$ (*i.e.* the signed subgraph of (G, σ) induced by the vertices of C and their neighbors which belong to $B_+ \cup B_-$), the integer k' and the two sets $B_+ \cap N_G(V(C))$ and $B_- \cap N_G(V(C))$.

Let \mathcal{CC}_{good} be the subset of \mathcal{CC} for which $\mathcal{I}'(C, 0)$ is a positive instance. We can determine \mathcal{CC}_{good} in $O(n^2)$ time as it suffices to check if $(G, \sigma)[V(C) \cup N_G(V(C))] \rightarrow_s H_{r,b}^{2-}$. The set $\mathcal{CC}_{bad} = \mathcal{CC} \setminus \mathcal{CC}_{good}$ is the set of instances for which at least one vertex must belong to a solution to \mathcal{I} . If $|\mathcal{CC}_{bad}| > k$, then we can reject as we cannot remove a vertex in each connected component of \mathcal{CC}_{bad} . We can safely ignore the connected components in \mathcal{CC}_{good} as they are already solved.

Note that $\mathcal{I}'(C, k')$ is an instance of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING if both $B_+ \cap N_G(V(C))$ and $B_- \cap N_G(V(C))$ are non-empty, and an instance CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING if exactly one of $B_+ \cap N_G(V(C))$ and $B_- \cap N_G(V(C))$ is empty. If both $B_+ \cap N_G(V(C))$ and $B_- \cap N_G(V(C))$ are empty, then $C \in \mathcal{CC}_{good}$ since $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$.

For $C \in \mathcal{CC}_{bad}$, let $\lambda(C)$ be the smallest integer k' for which $\mathcal{I}'(C, k')$ has a solution. Note that $\sum_{C \in \mathcal{CC}_{bad}} \lambda(C) \leq k$ if and only if our problem has a solution. Hence to solve our problem, it suffices, for every $C \in \mathcal{CC}_{bad}$, to compute $\lambda(C)$ which can be done in $O(\log(k))$ calls to a solver of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING or CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING by dichotomy. Hence in $O(k \log(k))$ calls to these algorithms, we can determine whether there is a solution to \mathcal{I} or not.

Moreover, we can avoid solving two instances of the form $\mathcal{I}'(C, k)$ and $\mathcal{I}'(C', k)$. Indeed, if we need to solve both, then this implies that $\mathcal{I}'(C, k-1)$ and $\mathcal{I}'(C', k-1)$ are negative instances. In particular, both problems need at least k vertices to be removed in their input signed graph and since $k \geq 1$, this means that we cannot solve both instances at the same time and we can reject. Therefore, at most one instance of the form $\mathcal{I}'(C, k)$ has to be solved by our algorithm.

Finally, remark that if \mathcal{I} is a nice instance then, for every $C \in \mathcal{CC}_{bad}$, both $B_+ \cap N_G(V(C))$ and $B_- \cap N_G(V(C))$ are non-empty. This implies that for every $k' \leq k$, $\mathcal{I}'(C, k')$ is an instance of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. In Figure 3.10, the connected components (G_1, σ_1) and (G_2, σ_2) induce instances of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING while (G_4, σ_4) induces an instance of CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING.

Lemma 3.30. *We have*

$$A(n, k) \leq O(n^2) + k \log(k) \max(\text{CA}(n, k-1), \text{CHA}(n, k-1)) \\ + \max(\text{CA}(n, k), \text{CHA}(n, k)),$$

and

$$A_{nice}(n, k) \leq O(n^2) + k \log(k) \text{CA}(n, k-1) + \text{CA}(n, k).$$

3.4.5 Separators: definitions and notation

In this section, we introduce the main concept that we use to find the solution.

Definition 3.31 (Separator). For every graph G and every two disjoint subsets of vertices A and B , an (A, B) -separator X is a set of vertices, disjoint from A and B , whose removal disconnects vertices of A from vertices of B . Let X, A and B be three sets of vertices of a graph G such that A and B are disjoint. The set $R(A, X)$ is the set of vertices of $G - X$ which are connected to any vertex of A . If X is an (A, B) -separator then let $R^{rest}(X)$ be the set of vertices of $G - X$ which are connected to none of the vertices of $A \cup B$. Note that if X is an (A, B) -separator, then $R(A, X)$ and $R(B, X)$ are disjoint.

Our goal is to find some separator of size $O(k)$ which intersects the solution to our problem. This way we can test, for every vertex of our separator, whether it belongs to the solution or not.

Definition 3.32 (Important (A, B) -separator). Let G be a graph, and let A and B be two disjoint sets of vertices. An *important* (A, B) -separator is an (A, B) -separator, minimal for inclusion, such that there is no (A, B) -separator S' verifying $|S'| \leq S$ and $R(A, S) \subsetneq R(A, S')$.

Theorem 3.33 ([45]). *The set $\mathcal{S}_k(A, B)$ of all important (A, B) -separators of size at most k , has size at most 4^k and can be constructed in $O(4^k k^2 n^2)$ time.*

In particular, the previous theorem implies that we can find an (A, B) -separator in $O(4^k k^2 n^2)$ time, if there is one. For more details, see [53].

3.4.6 Solving Connected Half Annotated Signed- $H_{r,b}^{2-}$ -Coloring

Let \mathcal{I} be an instance of CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of a signed graph (G, σ) , an integer k and a set of vertices B_+ such that $G - B_+$ is connected and $(G, \sigma) - B_+ \rightarrow_s H_{r,b}^{2-}$.

If $k < 0$ then we can reject \mathcal{I} . If $k = 0$, then it suffices to check whether $(G, \sigma) \rightarrow_s H_b^1$, which can be done in $O(n^2)$ time (see Theorem 2.8). Hence we can suppose $k > 0$.

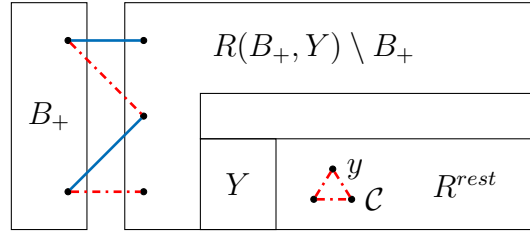


Figure 3.11: The structure of the input signed graph (G, σ) once an important (y, B_+) -separator Y has been found. Each polygon represent a subgraph of (G, σ) . The set $R(B_+, Y) \setminus B_+$ is the set of vertices of G which are not in B_+ and which are connected to a vertex of B_+ in $G - Y$. The set R^{rest} is the set of vertices which are disconnected from B_+ in $G - Y$. The signed subgraph of (G, σ) induced by R^{rest} contains an unbalanced cycle \mathcal{C} whose vertex set contains y .

If there is a solution S_1 to VERTEX DELETION H_b^1 -COLORING on input (G, σ) and integer k , then return S_1 . Suppose that there is no solution to VERTEX DELETION H_b^1 -COLORING on input (G, σ) and integer k , *i.e.* we cannot map (G, σ) to the positive loop by removing k vertices. Because of the previous assumption, it holds that $(G, \sigma) - B_+ \rightarrow_s H_r^1$ (as $(G, \sigma) - B_+$ is connected and $(G, \sigma) - B_+ \rightarrow_s H_{r,b}^{2-}$). Moreover if a solution S to \mathcal{I} exists, at least one connected component of $(G, \sigma) - S$ cannot be mapped to \oplus , otherwise VERTEX DELETION H_b^1 -COLORING would have a solution. This implies that S is a separator which separates the vertices of B_+ from the vertices of some unbalanced cycle \mathcal{C} . In particular, there exists $x \in V(\mathcal{C})$, such that S is an $(\{x\}, B_+)$ -separator.

If the set S exists, then there is an important $(\{x\}, B_+)$ -separator X of size at most k , possibly equal to S , for which $R(\{x\}, S) \subseteq R(\{x\}, X)$. In particular, the signed subgraph of (G, σ) induced by $R(\{x\}, X)$ contains the unbalanced cycle \mathcal{C} .

For every vertex $y \in V(G) \setminus B_+$, we construct the set $\mathcal{S}_k(\{y\}, B_+)$. This way, we construct at most $4^k n$ important separators in $O(4^k k^2 n^3)$ time. Recall that a signed graph (A, π) contains an unbalanced cycle if and only if it is not balanced, that is (A, π) is not equivalent to (A, \emptyset) , hence we can test the existence of an unbalanced cycle in a signed graph in $O(n^2)$ time by Theorem 2.8. If one important $(\{y\}, B_+)$ -separator Y , for some vertex $y \in V(G) \setminus B_+$, verifies that the signed subgraph of (G, σ) induced by $R(\{y\}, Y)$ contains an unbalanced cycle, then we keep this separator and discard the others. If no such separator exists, then X does not exist and we can reject.

Let Y be an important $(\{y\}, B_+)$ -separator, for a vertex $y \in V(G) \setminus B_+$, which verifies that the signed subgraph of (G, σ) induced by $R(\{y\}, Y)$ contains an unbalanced cycle (see Figure 3.11). If Y is a solution to \mathcal{I} , we can return Y . Otherwise, suppose that Y is not a solution.

Fix a solution S of \mathcal{I} and a homomorphism $\varphi_S : (G, \sigma) - S \rightarrow_s H_{r,b}^{2-}$ such that $\varphi(B_+) = \oplus$. The vertices of Y can belong to S , have image \oplus through φ or have image \ominus through φ . This separates Y into three sets. Set $Y_\cap^S = Y \cap S$, $Y_+^S = Y \cap \varphi^{-1}(\{\oplus\})$ and $Y_-^S = Y \cap \varphi^{-1}(\{\ominus\})$. For their part, the vertices of the solution S can belong to Y (and thus to Y_\cap), they can be connected to B_+ in $G - Y$, or they can be disconnected from B_+ in $G - Y$. Let $k_+^S = |S \cap R(B_+, Y)|$ be the number of vertices of S connected to B_+ in $G - Y$, let $R^{rest} = V(G) \setminus (Y \cup R(B_+, Y))$ be the vertices of $G - Y$ which are not connected to B_+ and $k_{rest}^S = |S \cap R_{rest}|$ be the number of vertices of S which belong to $G - Y$ and are not connected to B_+ . Our goal is to find the tuple $\theta_S = (Y_\cap^S, Y_+^S, Y_-^S, k_+^S, k_{rest}^S)$.

Let Θ be the set of all tuples of the form $(Y_\cap, Y_+, Y_-, k_+, k_{rest})$ where Y_\cap , Y_+ and Y_- form a partition of Y and k_+ and k_{rest} are two non-negative integers such that $k =$

$|Y_\cap| + k_+ + k_{rest}$. For $\theta = (Y_\cap, Y_+, Y_-, k_+, k_{rest}) \in \Theta$, we define the instance $\mathcal{I}_+(\theta)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of the signed subgraph $G_+(\theta)$ of (G, σ) induced by $R(B_+, Y) \cup (Y \setminus Y_\cap)$, the integer k_+ and the sets $B_+ \cup Y_+$ and Y_- . Similarly, we define the instance $\mathcal{I}_{rest}(\theta)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of the signed subgraph $G_{rest}(\theta)$ of (G, σ) induced by $R^{rest} \cup (Y \setminus Y_\cap)$, the integer k_{rest} and the sets Y_+ and Y_- .

Note that $S \cap R(B_+, Y)$ is a solution to the instance $\mathcal{I}_+(\theta_S)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING and that $S \cap R_{rest}$ is a solution to the instance $\mathcal{I}_{rest}(\theta_S)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING.

Moreover suppose that for some $\theta = (Y_\cap, Y_+, Y_-, k_+, k_{rest}) \in \Theta$, there exists a solution S_+ to $\mathcal{I}_+(\theta)$ and a solution S_{rest} to $\mathcal{I}_{rest}(\theta)$. Let φ_+ (resp. φ_{rest}) be a homomorphism from $G_+(\theta) - S_+$ (resp. $G_{rest}(\theta) - S_{rest}$) to $H_{r,b}^{2-}$ such that $\varphi_+(B_+ \cup Y_+) = \oplus$ (resp. $\varphi_{rest}(Y_+) = \oplus$) and $\varphi_+(Y_-) = \ominus$ (resp. $\varphi_{rest}(Y_-) = \ominus$). The homomorphism φ defined by $\varphi(x) = \varphi_+(x)$ if $x \in V(G_+(\theta))$ and $\varphi(x) = \varphi_{rest}(x)$ otherwise, verifies that $\varphi : (G, \sigma) - (S_+ \cup Y_\cap \cup S_{rest}) \rightarrow_s H_{r,b}^{2-}$ and $\varphi(B_+) = \oplus$. Indeed, if not, then there exist u and v in $V(G) \setminus (S_+ \cup Y_\cap \cup S_{rest})$ which are adjacent and such that $\varphi(u) = \oplus$ and $\varphi(v) = \ominus$. By definition of φ , one of u or v , say u , belongs to $V(G_+(\theta))$ and the other belongs to R_{rest} . Since no vertex of $R(B_+, Y)$ and R_{rest} are adjacent (recall that Y is a separator), $u \in Y$ and thus $u \in V(G_{rest})$. This contradicts the fact that S_{rest} is a solution of $\mathcal{I}_{rest}(\theta)$. Hence $S_+ \cup Y_\cap \cup S_{rest}$ is a solution to \mathcal{I} of size at most $k_+ + |Y_\cap| + k_{rest} = k$.

Hence to solve our problem, it suffices to find some $\theta \in \Theta$ for which the two instances $\mathcal{I}_+(\theta)$ and $\mathcal{I}_{rest}(\theta)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING have a solution. If no such θ exists then θ_S does not exist and we can reject.

In order to find this θ , we will try every possibility. There are at most 3^k ways to partition Y into three sets, there are at most $k+1$ choices for k_+ and once k_+ and Y_\cap are chosen, we only have one choice for k_{rest} . Hence we will make at most $2 \cdot 3^k(k+1)$ calls to a solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING.

In most calls to a solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING, the instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING has a parameter (k_+ or k_{rest}) smaller than k . In some cases, the parameter does not decrease.

The first of those cases is when $k_{rest} = k$. Since $(G, \sigma) - B_+ \rightarrow_s H_{r,b}^{2-}$, any set is a solution to $\mathcal{I}_{rest}(\theta)$ for any θ of the form $(\emptyset, Y_+, Y_-, 0, k)$. As Y is not a solution, the empty set is not a solution of $\mathcal{I}_+(\theta)$ for any θ of the form $(\emptyset, Y_+, Y_-, 0, k)$, *i.e.* a solution S must intersect the signed subgraph of (G, σ) induced by $R(B_+, Y)$. Hence any $\theta \in \Theta$ with $k_{rest} = k$ does not yield a solution, hence we can ignore these θ 's when trying all possibilities.

The other case is when $k_+ = k$. We claim that the only interesting $\theta \in \Theta$ to test with $k_+ = k$ is $(\emptyset, \emptyset, Y, k, 0)$. Indeed, recall that the signed subgraph G' of (G, σ) induced by $Y \cup R_{rest}$ contains an unbalanced cycle. Moreover G' is connected as G is connected and Y is an important separator, hence every vertex of G' must be mapped to \ominus . In particular this is the case for the vertices of Y . This implies $Y_+ = \emptyset$ and $Y_- = Y$.

Moreover, for $\theta = (\emptyset, \emptyset, Y, k, 0)$, the instance $\mathcal{I}_+(\theta)$ is actually a nice instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. Indeed let u be a vertex of $V(G_+(\theta)) \setminus (Y \cup B_+) = R(B_+, Y) \setminus B_+$, u is connected to a vertex of B_+ by a path with internal vertices in $R(B_+, Y)$ by definition of $R(B_+, Y)$. Since $G - B_+$ is connected, there is a shortest path between u and a vertex of Y in $G - B_+$, this path has internal vertices in $R(B_+, Y)$ by definition of Y , hence it is also a path in $G_+(\theta)$.

Let us summarize. We make at most $2 \cdot 3^k(k+1)$ calls to a solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING on a instance with a smaller parameter. Hence it takes a $O(3^k(k+1)A(n, k-1))$ time to perform these calls. On top of these, we perform at most one call to the solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING on a nice instance with parameter k . Hence it takes a time $A_{nice}(n, k)$ to perform this call.

Lemma 3.34. *We have:*

$$\text{CHA}(n, k) \leq 2^{O(k)}n^{O(1)} + 2 \cdot 3^k(k+1)A(n, k-1) + A_{nice}(n, k).$$

3.4.7 Solving Connected Annotated Signed- $H_{r,b}^{2-}$ -Coloring

Let \mathcal{I} be an instance of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of a signed graph (G, σ) of order n , an integer k and two disjoint non-empty sets of vertices B_+ and B_- such that $G - (B_+ \cup B_-)$ is connected and $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$. In order to solve CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING, we need to introduce some notation.

A subset of vertices A of $V(G)$ is *big* if $|A| \geq \frac{9}{10}n$. A subset of vertices A of $V(G)$ is *small* if $|A| \leq \frac{1}{10}n$. Note that in a partition of $V(G)$, there is at most one big part and if there is a big part, then all the other parts are small.

The main idea of the resolution of CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING is to find some “good” (B_+, B_-) -separator in G . The following lemma realizes just that.

Lemma 3.35. *Let G be a graph of order n , k be an integer and A and B be two disjoint sets of vertices such that $G - (A \cup B)$ is connected. We can find in $f(k)n^d$ time one of the following:*

1. an (A, B) -separator X of size at most k such that $R(A, X)$ and $R(B, X)$ are not big, or
2. two (A, B) -separators X and Y of size at most k such that $R(B, X)$ is big, $R(A, Y)$ is big and the graph $G[R^{rest}(X \cup Y) \cup X \cup Y]$ contains at least $k+1$ paths P_1, \dots, P_{k+1} , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for $u_i \in X$ and $v_i \in Y$, or
3. an (A, B) -separator X (resp. Y) of size at most k such that $R(B, X)$ (resp. $R(A, Y)$) is big, and the graph $G[R(B, X) \cup X]$ (resp. $G[R(A, Y) \cup Y]$) contains at least $k+1$ paths P_1, \dots, P_{k+1} , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for $u_i \in X$ and $v_i \in B$ (resp. $u_i \in A$ and $v_i \in Y$), or
4. two (A, B) -separators X and Y of size at most k such that $R(A, X)$ and $R(B, Y)$ are small and there is a (u, v) -path, $u \in A$ and $v \in B$, which contains only vertices in $R(A, X) \cup X \cup Y \cup R(B, Y)$, or
5. an (A, B) -separator X (resp. Y) of size at most k such that $R(B, X)$ (resp. $R(A, Y)$) is big and there exists a (u, v) -path for two vertices $u \in A$ and $v \in B$ whose internal vertices are contained in $R(A, X) \cup X$ (resp. $R(B, Y) \cup Y$), or
6. we can determine that there exists no (A, B) -separator X of size at most k .

See Figure 3.12 for a schematic view of the first five cases.

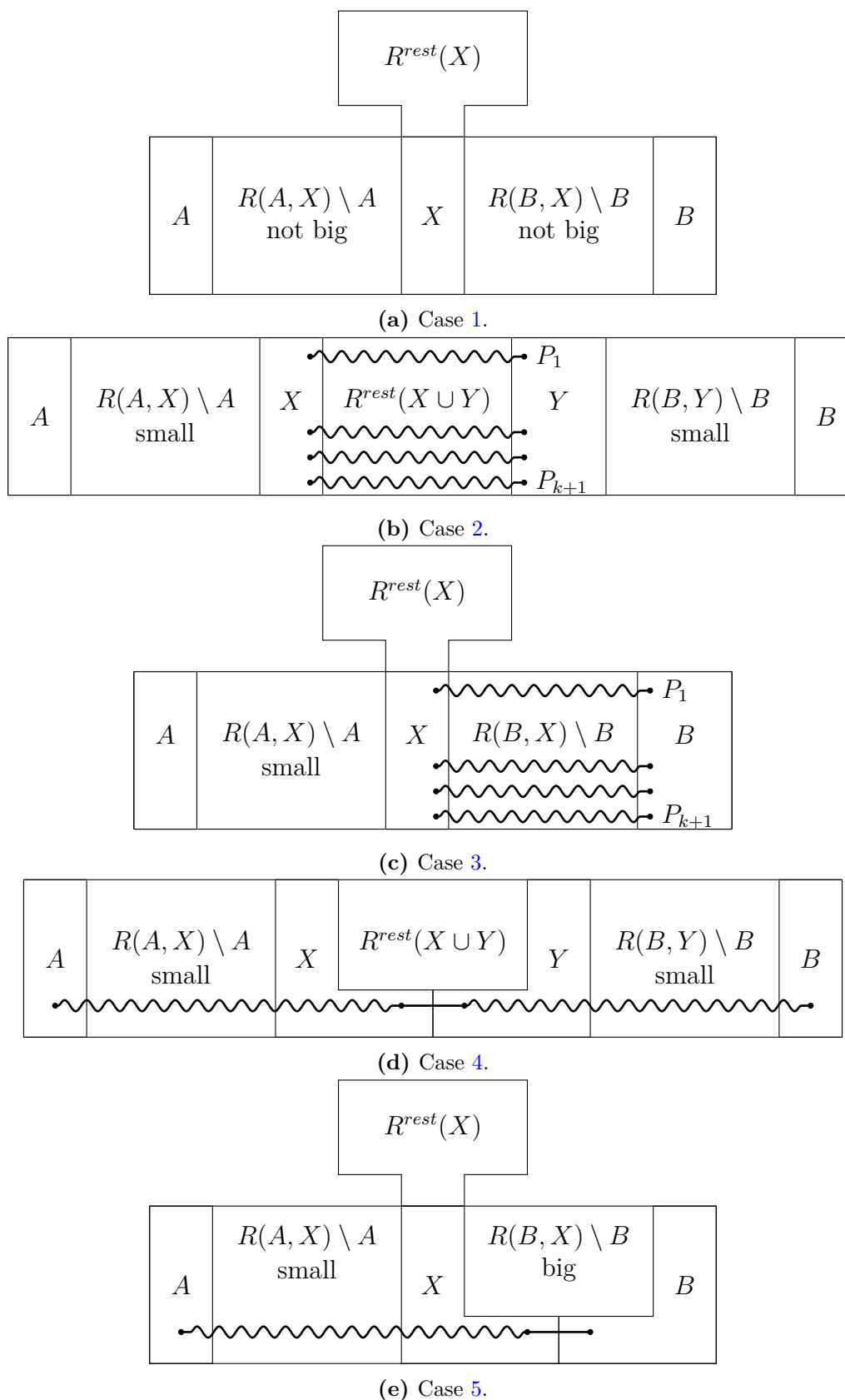


Figure 3.12: The first five cases of Lemma 3.35. Each polygon represent a subgraph of G . Wavy lines between two vertices represent arbitrary long paths between these two vertices.

Note that every solution S to \mathcal{I} must disconnect the vertices of B_+ from the vertices of B_- as they are mapped to disconnected vertices of $H_{r,b}^{2-}$. If $n < 100k$, then we can solve the problem by a brute force approach. Assume that $n > 100k$. We start by applying Lemma 3.35 for $A = B_+$ and $B = B_-$. If we are in case 6, then S does not exist and we can reject.

Let X be an (B_+, B_-) -separator. Up to modifying X , we can suppose that X is minimal for inclusion. Let $\Theta(X)$ be the set of all tuples of the form $(X_\cap, X_+, X_-, k_+, k_-, k_{rest})$ where X_\cap, X_+ and X_- are a partition of X and k_+, k_- and k_{rest} are three non-negative integers such that $|X_\cap| + k_+ + k_- + k_{rest} = k$. If $|X| \leq k$ then we remove every θ for which $k_{rest} = k$ from $\Theta(X)$.

For $\theta \in \Theta(X)$, let $\mathcal{I}^+(\theta)$ (resp. $\mathcal{I}^-(\theta)$, resp. $\mathcal{I}^{rest}(\theta)$) be the instance of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING composed of the signed graph $(G, \sigma)[R(B_+, X) \cup (X \setminus X_\cap)]$ (resp. $(G, \sigma)[R(B_-, X) \cup (X \setminus X_\cap)]$, resp. $(G, \sigma)[R^{rest}(X) \cup (X \setminus X_\cap)]$), the integer k_+ (resp. k_- , resp. k_{rest}) and the two sets $B_+ \cup X_+$ (resp. X_+ , resp. X_+) and X_- (resp. $B_- \cup X_-$, resp. X_-).

Let S be a solution to \mathcal{I} and $\varphi_S : (G, \sigma) - S \rightarrow_s H_{r,b}^{2-}$ for which $\varphi_S(B_+) = \{\oplus\}$ and $\varphi_S(B_-) = \{\ominus\}$. Let $X_+^S = \varphi_S^{-1}(\{\oplus\}) \cap X$, $X_-^S = \varphi_S^{-1}(\{\ominus\}) \cap X$ and $\theta_S = (X \cap S, X_+^S, X_-^S, |S \cap R(B_+, X)|, |S \cap R(B_-, X)|, |S \cap R^{rest}(X)|)$.

Let us highlight a few facts. First if $|S \cap R^{rest}(X)| = k$ and $|X| \leq k$, then $S \subseteq R^{rest}(X)$ and X is a solution since $(G, \sigma) - (B_+ \cup B_-) \rightarrow_s H_{r,b}^{2-}$. Hence, if X is not a solution and has size at most k , then we do not need to test the cases for which $k_{rest} = k$. Moreover, $S \cap R(B_+, X)$ (resp. $S \cap R(B_-, X)$, resp. $S \cap R^{rest}(X)$) is a solution to $\mathcal{I}^+(\theta_S)$ (resp. $\mathcal{I}^-(\theta_S)$, resp. $\mathcal{I}^{rest}(\theta_S)$). Hence, if S exists, then X is a solution or there exists $\theta \in \Theta(X)$ for which $\mathcal{I}^+(\theta)$, $\mathcal{I}^-(\theta)$ and $\mathcal{I}^{rest}(\theta)$ are positive instances of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. Finally, if S^+ is a solution to $\mathcal{I}^+(\theta)$, S^- is a solution to $\mathcal{I}^-(\theta)$ and S^{rest} is a solution to $\mathcal{I}^{rest}(\theta)$ for some $\theta = (X_\cap, X_+, X_-, k_+, k_-, k_{rest}) \in \Theta(X)$, then $X_\cap \cup S^+ \cup S^- \cup S^{rest}$ is a solution to \mathcal{I} since the three problems intersect only on X and we fixed the homomorphism on X with the choice of X_+ and X_- .

Hence to solve our problem, it suffices to find some $\theta \in \Theta(X)$ for which $\mathcal{I}^+(\theta)$, $\mathcal{I}^-(\theta)$ and $\mathcal{I}^{rest}(\theta)$ are positive instances of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. Finally, note that if $|X| \leq 2k$, then $|\Theta(X)| \leq 3^{2k}(k+1)^2$. Let us note $g(k) = 3^{2k}(k+1)^2$.

We can now treat the other cases of Lemma 3.35.

1. Suppose that we have found a minimal (B_+, B_-) -separator X of size at most k such that $R(B_+, X)$ and $R(B_-, X)$ are not big. If X is a solution, then return X . For each $\theta \in \Theta(X)$, we solve the instances $\mathcal{I}^+(\theta)$, $\mathcal{I}^-(\theta)$ and $\mathcal{I}^{rest}(\theta)$ of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING. If the three instances are positive instances, then we accept. If no θ yields to our algorithm accepting, then we reject. In most cases, the parameters $(k_+, k_-$ and $k_{rest})$ are all smaller than k , hence the three sub-problems are solved in time at most $3A(n, k-1)$. The case $k_{rest} = k$ does not occur, by construction of $\Theta(X)$ as X is not a solution.

The case where $k_+ = k$, implies $\theta = (\emptyset, \emptyset, X, k, 0, 0)$ as every vertex of X is connected to B_- (and we cannot disconnect them since $k_- = 0$). Moreover, in this case, $\mathcal{I}^-(\theta)$ and $\mathcal{I}^{rest}(\theta)$ can be solved in polynomial time as k_- and k_{rest} are equal to 0. We can start by solving these two instances, and if one of the two instances is not a positive instance, then we do not have to solve $\mathcal{I}^+(\theta)$ to rule out this θ .

Similarly, the case $k_- = k$, implies $\theta = (\emptyset, X, \emptyset, 0, k, 0)$. Note that it is not possible to solve both $\mathcal{I}^+(\emptyset, \emptyset, X, k, 0, 0)$ and $\mathcal{I}^-(\emptyset, X, \emptyset, 0, k, 0)$ as this would imply that X is a solution. Indeed if we call both instances then this means that $(G, \sigma)[R(B_+, X) \cup X] \rightarrow_s H_b^1$, $(G, \sigma)[R(B_-, X) \cup X] \rightarrow_s H_r^1$, and $(G, \sigma)[R^{rest}(X) \cup X] \rightarrow_s H_{r,b}^{2-}$ and thus $(G, \sigma) - X \rightarrow_s H_{r,b}^{2-}$.

Hence at most one resolved sub-problem has its parameter equal to k . Recall that by the choice of X , $|R(B_+, X)| \leq \frac{9}{10}n$ and $|R(B_-, X)| \leq \frac{9}{10}n$. This implies that the instance with parameter equal to k can be solved in $A(\frac{91}{100}n, k)$ time (recall that $k < \frac{n}{100}$).

2. Suppose that we have found two (B_+, B_-) -separators X and Y of size at most k such that $R(B_-, X)$ is big, $R(B_+, Y)$ is big and the graph $G[R^{rest}(X \cup Y) \cup X \cup Y]$ contains at least $k + 1$ paths P_1, \dots, P_{k+1} , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for $u_i \in X$ and $v_i \in Y$. If one of X or Y is a solution then we return it. We consider the (B_+, B_-) -separator $X \cup Y$. Since $|X \cup Y| > k$, $X \cup Y$ is not a solution thus it is possible to have to test some θ with $k_{rest} = k$. Since $G[R^{rest}(X \cup Y) \cup X \cup Y]$ contains $k + 1$ paths (with disjoint internal vertices) from B_+ to B_- , a solution S cannot be contained in $R^{rest}(X \cup Y) \cup X \cup Y$, as it would not be a (B_+, B_-) -separator. Hence, we do not have to test any $\theta \in \Theta(X \cup Y)$ with $k_{rest} = k$. As in the previous case, most sub-problems have a parameter smaller than k .

The case where $k_+ = k$, implies $\theta = (\emptyset, \emptyset, X \cup Y, k, 0, 0)$ as every vertex of $X \cup Y$ is connected to B_- (and we cannot disconnect them since $k_- = 0$ and $k_{rest} = 0$). Moreover, in this case, $\mathcal{I}^-(\theta)$ and $\mathcal{I}^{rest}(\theta)$ can be solve in polynomial time as k_- and k_{rest} are equal to 0. We can start by solving these two instances, and if one of the two instances is not a positive instance, then we do not have to solve $\mathcal{I}^+(\theta)$ to rule out this θ .

Similarly, the case $k_- = k$, implies $\theta = (\emptyset, X \cup Y, \emptyset, 0, k, 0)$. We can apply the same argument as before: if we call both $\mathcal{I}^+(\emptyset, \emptyset, X \cup Y, k, 0, 0)$ and $\mathcal{I}^-(\emptyset, X \cup Y, \emptyset, 0, k, 0)$ then this means that both X and Y are solutions to our problem.

Hence we need to resolve at most one sub-problem having its parameter equal to k . Recall that by the choice of X and Y , $|R(B_+, X)| \leq \frac{1}{10}n$ and $|R(B_-, Y)| \leq \frac{1}{10}n$ (as $R(B_-, X)$ and $R(B_+, Y)$ are big). This implies that the instance with parameter equal to k can be solved in $A(\frac{11}{100}n, k)$ time (recall that $k < \frac{n}{100}$).

3. Suppose that we have found an (B_+, B_-) -separator X of size at most k such that $R(B_-, X)$ is big, and the graph $G[R(B_-, X) \cup X]$ contains at least $k + 1$ paths P_1, \dots, P_{k+1} , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for $u_i \in X$ and $v_i \in B_-$. This case is very similar to the previous case. The subgraph $G[R(B_-, X) \cup X]$ of G cannot entirely contain a solution S due to the existence of the $k + 1$ paths P_1, \dots, P_{k+1} . Hence, we do not have to test any $\theta \in \Theta(X)$ with $k_- = k$. Since $R(B_-, X)$ is big, $R(B_+, X)$ and $R^{rest}(X)$ are small. The only θ for which $k_+ = k$ is $(\emptyset, \emptyset, X, k, 0, 0)$ and if $k_{rest} = k$, then X is a solution. Hence we only have one sub-problem with parameter equal to k , and it can be solved in $A(\frac{11}{100}n, k)$ time.
4. Suppose that we have found two (B_+, B_-) -separators X and Y of size at most k such that $R(B_+, X)$ and $R(B_-, Y)$ are small and there is a (u, v) -path P , $u \in B_+$

and $v \in B_-$, which contains only vertices in $R(B_+, X) \cup X \cup Y \cup R(B_-, Y)$. This case is very similar to the second case. If one of X or Y is a solution, then return it. We will test every $\theta \in \Theta(X \cup Y)$ to find our solution. Note that a solution S cannot be included in $R^{rest}(X \cup Y)$ as it would not intersect P , hence we can suppose that $k_{rest} \neq k$. The rest of the analysis is the same as for the second sub-case.

5. Suppose that we have found a (B_+, B_-) -separator X of size at most k such that $R(B_-, X)$ is big and there exists a (u, v) -path P for two vertices $u \in B_+$ and $v \in B_-$ whose internal vertices are contained in $R(B_+, X) \cup X$. As before, since $|X| \leq k$, we do not have to test any θ with $k_{rest} = k$. Due to the existence of P , any solution S must intersect $R(B_+, X) \cup X$ and cannot be entirely contained in $R(B_-, X)$. Hence, we do not have to test any θ with $k_- = k$. The only case where the parameter is k , is when $k_+ = k$ and $\theta = (\emptyset, \emptyset, X, k, 0, 0)$ which we can solve in $A(\frac{91}{100}n, k)$ time.

To conclude in every case, after applying Lemma 3.35, we can solve our problem in at most $3g(k)$ calls to a solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING on an instance where k decreases and at most one call to a solver of ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING on an instance where the order of the input graph is at most $\frac{91}{100}n$.

Lemma 3.36. *We have:*

$$CA(n, k) \leq f(k)n^d + 3g(k)A(n, k-1) + A\left(\frac{91}{100}n, k\right).$$

3.4.7.1 Proof of Lemma 3.35

This section is dedicated to the proof of Lemma 3.35. In this proof, we always suppose that our separators are minimal since we can transform a separator into a minimal separator in polynomial time.

Let G be a graph of order n , k be an integer and A and B be two disjoint sets of vertices such that $G - (A \cup B)$ is connected. If no (A, B) -separator of size at most k exists then we are in case 6. Suppose otherwise and let $G_0 = G$, $X_0 = A$ and $Y_0 = B$. Let us find a minimal (X_0, Y_0) -separator S_0 of size at most k in G_0 . If S_0 verifies 1, *i.e.* none of $R(X_0, S_0)$ or $R(Y_0, S_0)$ is big, then we can conclude. W.l.o.g assume that $R(X_0, S_0)$ is big. Let G_1 be the graph $G_0[R(X_0, S_0) \cup S_0]$ and set $X_1 = X_0$ and $Y_1 = S_0$.

Suppose that we have constructed G_i and two disjoint sets of vertices X_i and Y_i of G_i . Since we work with several graphs, we note $R_i(S, T)$ for the set $R(S, T)$ taken in the graph G_i . The notation $R(S, T)$ always refer to $R_0(S, T)$. If no (X_i, Y_i) -separator of size at most k exists, then we stop. Find a minimal (X_i, Y_i) -separator S_i of size at most k in G_i . If none of $R(A, S_i)$ or $R(B, S_i)$ is big, then we also stop. If $R(A, S_i)$ is big, then set $G_{i+1} = G_i[R_i(X_i, S_i) \cup S_i]$, $X_{i+1} = X_i$ and $Y_{i+1} = S_i$. If $R_i(B, S_i)$ is big, then set $G_{i+1} = G_i[R_i(Y_i, S_i) \cup S_i]$, $X_{i+1} = S_i$ and $Y_{i+1} = Y_i$. As G_{i+1} has less vertices than G_i , this process stops in at most n steps. See Figure 3.13 for an example of construction of G_{i+1} from G_i .

Suppose that the process stops for G_p and the two sets X_p and Y_p where $p \geq 1$ (see Figure 3.14). Let us see one important property of our construction.

Claim 3.37. *For every $i \leq p-1$, the minimal (X_{i+1}, Y_{i+1}) -separator S_{i+1} in G_{i+1} is also a minimal (X_i, Y_i) -separator in G_i .*

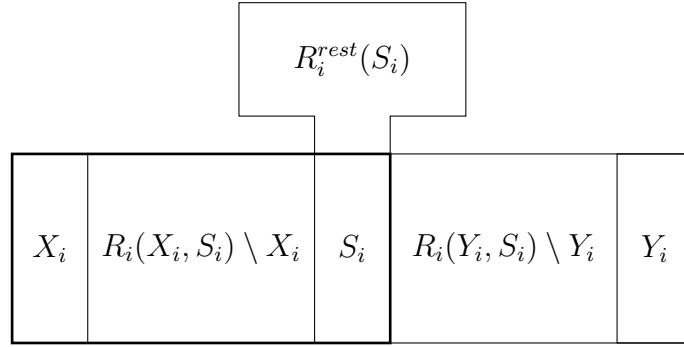


Figure 3.13: Some graph G_i constructed in the algorithm for Lemma 3.35. In this case, we suppose that $R(A, S_i)$ is big. The bold rectangle highlights the graph G_{i+1} .

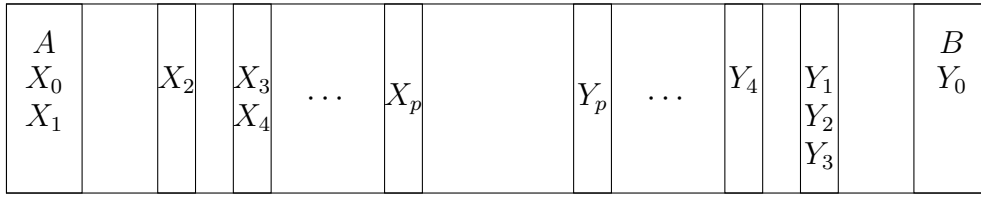


Figure 3.14: An example of execution of the algorithm yielding to Lemma 3.35. In this execution, we chose $Y_1 = S_0$, $X_2 = S_1$, $X_3 = S_2$ and $Y_4 = S_3$.

Proof. By symmetry, suppose that $R(A, S_i)$ is big and thus $X_{i+1} = X_i$ and $Y_{i+1} = S_i$.

Let $u \in X_i$ and $v \in Y_i$ be two vertices of G_i and let P be a (u, v) -path P . By definition of S_i , P intersect S_i . Therefore P is the concatenation of two paths P^1 and P^2 such that P^1 is a (u, w) -path contained in G_{i+1} where $w \in S_i$. By definition of S_{i+1} , P^1 intersects S_{i+1} which implies that P intersects S_{i+1} . Hence S_{i+1} is an (X_i, Y_i) -separator.

Suppose, by contradiction that S_{i+1} is not a minimal (X_i, Y_i) -separator. This implies that there exists $s \in S_{i+1}$ such that $S_{i+1} \setminus \{s\}$ is also an (X_i, Y_i) -separator. Note that since S_{i+1} is a minimal (X_i, S_i) -separator, there is a path P_1 from s to a vertex x of X_i whose internal vertices are in $R_{i+1}(X_i, S_{i+1}) \subseteq R_i(X_i, S_{i+1})$. If there is an (s, y) -path P_2 for some $y \in Y_i$ whose internal vertices are in $R_i(Y_i, S_{i+1})$, then the path P obtained by concatenating P_1 and P_2 is a path from X_i to Y_i not intersecting $S_{i+1} \setminus \{s\}$, a contradiction. By minimality of S_{i+1} , there is an (s, w) -path P_3 whose internal vertices belong to $R_{i+1}(S_i, S_{i+1})$ and for which $w \in S_i$. By minimality of S_i , there is a (w, y) -path P_4 whose internal vertices belong to $R_i(Y_i, S_i)$ and for which $y \in Y_i$. Note that $R_i(Y_i, S_i) \cap V(G_{i+1}) = \emptyset$, hence S_{i+1} does not intersect P_3 nor P_4 . Hence the concatenation of P_3 and P_4 is a path from s to some vertex y of Y_p whose internal vertices do not intersect S_{i+1} , they thus belong to $R_i(Y_i, S_{i+1})$, a contradiction. \square

In particular, Claim 3.37 proves that every S_i is an (A, B) -separator.

There are multiple cases which can make us stop. Recall that, by our assumption on S_0 , we have $Y_p \neq B$. We first distinguish between the case where $X_p = A$ and the case where $X_p \neq A$. When treating a case, we suppose that all previous cases do not occur.

Suppose that $X_p = A$. Since $X_p = A$, for every $i \leq p-1$, $R(A, S_i)$ is big and $Y_{i+1} = S_i$. Recall that Y_p is a minimal (A, B) -separator by Claim 3.37.

- Suppose that there is no (A, Y_p) -separator of any size in G_p . This implies that there is an edge xy between a vertex x of A and a vertex y of Y_p . Since Y_p is a minimal

(A, B) -separator, there is a path from y to a vertex b of B whose internal vertices belong to $R(B, Y_p)$. By concatenating this path with the edge xy , we obtain an (x, b) -path whose internal vertices belong to $R(B, Y_p) \cup Y_p$. Hence Y_p verifies 5.

- *Suppose that there is no (A, Y_p) -separator of size at most k in G_p .* We want to show that Y_p verifies 3. Note that by construction $R(A, Y_p)$ is big. Note that, $G_p = G[R(A, Y_p) \cup Y_p]$. As there is no (A, Y_p) -separator of size at most k in G_p , there exist at least $k + 1$ paths P_1, \dots, P_{k+1} in G_p , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for some $u_i \in A$ and $v_i \in Y_p$. Hence Y_p verifies 3.
- *Suppose that S_p exists and none of $R(A, S_p)$ or $R(B, S_p)$ is big.* By Claim 3.37, S_p is an (A, B) -separator, hence 1 holds.

Suppose that $X_p \neq A$. Recall that both X_p and Y_p are minimal (A, B) -separators by Claim 3.37. Note also that by construction, one of X_p or Y_p is the minimal (A, B) -separator S_{p-1} . In particular one of $R(A, S_{p-1})$ or $R(B, S_{p-1})$ is big. By symmetry, suppose w.l.o.g. that $R(A, S_{p-1})$ is big and $S_{p-1} = Y_p$. In particular, $R(B, Y_p)$ is small.

- *Suppose that there is no (X_p, Y_p) -separator of any size in G_p .* This implies that there is an edge xy between a vertex x of X_p and a vertex y of Y_p . By construction of the algorithm, X_p is some S_{i_x} , $i_x \leq p - 1$, for which $R(B, S_{i_x})$ is big. As $R(B, X_p)$ is big, $R(A, X_p)$ is small. As X_p and Y_p are minimal (A, B) -separators, there exists a (u, x) -path for some $u \in A$ which contains only vertices in $R(A, X_p) \cup X_p$ and a (y, v) -path for some $v \in B$ which contains only vertices in $R(B, Y_p) \cup Y_p$. By concatenating these two paths, we obtain a (u, v) -path containing only vertices in $R(A, X_p) \cup X_p \cup R(B, Y_p) \cup Y_p$. As $Y_p \cap R(A, X_p) = \emptyset$ and $X_p \cap R(B, Y_p) = \emptyset$, this concatenated path contains only vertices in $R(A, X_p \cup Y_p) \cup X_p \cup Y_p \cup R(B, X_p \cup Y_p)$, which implies 4.
- *Suppose that there is no (X_p, Y_p) -separator of size at most k in G_p .* We want to show that X_p and Y_p verify 2. Note that by construction $R(A, Y_p)$ is big. By construction, X_p is some S_{i_x} , $i_x \leq p - 1$, for which $R(B, S_{i_x})$ is big. Note that, $V(G_p) \subseteq R(B, X_p)$ and $V(G_p) \subseteq R(A, Y_p)$. This implies that in $G - X_p - Y_p$, u is not connected to A nor to B , hence $V(G_p) \subseteq R^{rest}(X_p \cup Y_p) \cup X_p \cup Y_p$. As there is no (X_p, Y_p) -separator of size at most k in G_p , there exist at least $k + 1$ paths P_1, \dots, P_{k+1} , with disjoint internal vertices, such that each P_i is a (u_i, v_i) -path for $u_i \in X_p$ and $v_i \in Y_p$. Hence X_p and Y_p verify 2.
- *Suppose that S_p exists and none of $R(A, S_p)$ or $R(B, S_p)$ is big.* By Claim 3.37, S_p is an (A, B) -separator, hence 1 holds.

Complexity. Let us conclude with some complexity analysis. It takes an FPT time to find a separator of size at most k . We repeat this step at most n times. Moreover, each stop conditions can be checked in polynomial time. Hence, we find our separator in a $f(k)n^d$ time for some computable function f and some fixed integer d .

3.4.8 Complexity analysis

We recall that we have obtained the following bounds:

$$\begin{aligned}
 D(n, k) &\leq 2^{k+1}(O(k^2) + A(n, k)), \\
 A(n, k) &\leq O(n^2) + k \log(k) \max(CA(n, k-1), \text{CHA}(n, k-1)) \\
 &\quad + \max(CA(n, k), \text{CHA}(n, k)), \\
 A_{\text{nice}}(n, k) &\leq O(n^2) + k \log(k)CA(n, k-1) + CA(n, k), \\
 CA(n, k) &\leq f(k)n^d + 3g(k)A(n, k-1) + A\left(\frac{91}{100}n, k\right), \text{ and} \\
 \text{CHA}(n, k) &\leq 2^{O(k)}n^{O(1)} + 2 \cdot 3^k(k+1)A(n, k-1) + A_{\text{nice}}(n, k).
 \end{aligned}$$

First note that $CA(n, k) \leq \text{CHA}(n, k) \leq A(n, k)$, hence, we can express $A(n, k)$ as follows.

$$\begin{aligned}
 A(n, k) &\leq O(n^2) + k \log(k)\text{CHA}(n, k-1) + \text{CHA}(n, k), \\
 &\leq O(n^2) + k \log(k) \left(2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-2) + A_{\text{nice}}(n, k-1)\right) \\
 &\quad + \left(2^{O(k)}n^{O(1)} + 2 \cdot 3^k(k+1)A(n, k-1) + A_{\text{nice}}(n, k)\right), \\
 &\leq 2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-1) + A_{\text{nice}}(n, k), \\
 &\leq 2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-1) + O(n^2) + k \log(k)CA(n, k-1) + CA(n, k), \\
 &\leq 2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-1) + CA(n, k).
 \end{aligned}$$

With the upper bound on $CA(n, k)$ and since $f(k) = 2^{O(k)}$ and $g(k) = 2^{O(k)}$, we can simplify the previous identity as follows.

$$\begin{aligned}
 A(n, k) &\leq 2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-1) \\
 &\quad + f(k)n^d + 3g(k)A(n, k-1) + A\left(\frac{91}{100}n, k\right), \\
 &\leq 2^{O(k)}n^{O(1)} + 2^{O(k)}A(n, k-1) + A\left(\frac{91}{100}n, k\right).
 \end{aligned}$$

Hence for some integers λ , μ and c , we have:

$$A(n, k) \leq 2^{\lambda k}n^c + 2^{\mu k}A(n, k-1) + A\left(\frac{91}{100}n, k\right).$$

Let $x > 0$ be a real number such that $\left(\frac{100}{91}\right)^x = n$. We have $x = \frac{\log(n)}{\log(100/91)}$. Note that $x \leq 25 \log(n)$. Let us expand the last term of the sum.

$$\begin{aligned}
 A(n, k) &\leq \sum_{i=0}^{25 \log(n)} \left(2^{\lambda k} \left(\frac{91}{100} \right)^{ic} n^c + 2^{\mu k} A \left(n \left(\frac{91}{100} \right)^i, k-1 \right) \right), \\
 &\leq 25 \log(n) \left(2^{\lambda k} n^c + 2^{\mu k} A(n, k-1) \right), \\
 &\leq 25 \cdot 2^{\lambda k} n^c \log(n) + 25 \log(n) 2^{\mu k} A(n, k-1), \\
 &\leq 25 \cdot 2^{\lambda k} n^c \log(n) \times \sum_{i=0}^k \left(25 \log(n) 2^{\mu k} \right)^i, \\
 &\leq 25 \cdot 2^{\lambda k} n^c \log(n) \times k \left(25 \log(n) 2^{\mu k} \right)^k, \\
 &\leq 2^{O(k^2)} n^{O(1)} \log(n)^k.
 \end{aligned}$$

As $\log(n)^k = O(n)$, we obtain:

$$A(n, k) \leq 2^{O(k^2)} n^{O(1)}.$$

This implies $D(n, k) \leq 2^{O(k^2)} n^{O(1)}$ and proves Theorem 3.26.6 since we can solve our problem in $O(2^k n) D(n, k)$ time.











3.5 Edge Deletion Signed- (H, π) -Coloring




As for the vertex deletion case, this problem is only interesting when (H, π) is an s-core.

Among these problems, the problem EDGE DELETION SIGNED- H_b^1 -COLORING corresponds to computing the minimum number of edges which need to be removed from the input in order to make the input signed graph balanced. This problem is exactly the problem of determining the frustration index of the input. Since being introduced by Harary [93] in 1959, this notion has been extensively studied (*e.g.* see [67, 94, 111, 149]). Computing the frustration index of a signed graph is NP-hard [111]. We complete the result by showing that this problem is FPT.

The following theorem solves EDGE DELETION SIGNED- (H, π) -COLORING for all cases.

Theorem 3.38. *Let (H, π) be an s-core. The following statements hold.*

1. *The problem EDGE DELETION SIGNED- H_{rb}^1 -COLORING and the problem EDGE DELETION SIGNED- H_-^1 -COLORING are in P.*
2. *The problem EDGE DELETION SIGNED- (H, π) -COLORING is NP-complete even for $k = 0$ when (H, π) has at least three edges (see Theorem 2.22).*
3. *The problem EDGE DELETION SIGNED- (H, π) -COLORING is NP-complete when (H, π) is one of H_b^1 () , $H_{-,-}^{2b}$ () , $H_{-,-}^{2r}$ () , $H_{-,-}^{2rb}$ () or $H_{r,b}^{2-}$ () .*
4. *The problem EDGE DELETION SIGNED- (H, π) -COLORING is FPT when (H, π) is one of H_b^1 () , H_r^1 () , $H_{-,-}^{2b}$ () , $H_{-,-}^{2rb}$ () or $H_{r,b}^{2-}$ () .*

Proof. The problem EDGE DELETION SIGNED- H_{rb}^1 -COLORING is in P as every signed graph admits a homomorphism to H_{rb}^1 (). The problem EDGE DELETION SIGNED- H_-^1 -COLORING is equivalent to counting the number of edges of the input graph, hence it is in P. By Theorem 2.22, if (H, π) has at least three edges then EDGE DELETION SIGNED- (H, π) -COLORING is NP-complete even for $k = 0$. Note that for every (H, π) , EDGE DELETION SIGNED- (H, π) -COLORING is in NP. By Observation 2.14, the complexity for H_r^1 () is the same as for H_b^1 (). The problem EDGE DELETION SIGNED- $H_{-, -}^{2rb}$ -COLORING is exactly the problem EDGE BIPARTIZATION, hence it is NP-complete and FPT. The problem EDGE DELETION SIGNED- H_b^1 -COLORING is equivalent to computing the frustration index of the input graph which is NP-hard [111]. It is also equivalent to EDGE DELETION H -COLORING where H is the 2-edge-colored graph isomorphic to the signed graph $\text{DSG}(H_b^1)$, hence the problem is FPT by Theorem 3.20. The problem EDGE DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING is equivalent to the problem EDGE BIPARTIZATION on input signed graphs which are all-positive, and is hence NP-complete.

We reduce EDGE DELETION SIGNED- H_b^1 -COLORING to EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING in order to show that EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING is NP-hard. Let $((G, \sigma), k)$ be an instance of EDGE DELETION SIGNED- H_b^1 -COLORING. We denote by x_1, \dots, x_n the vertices of G . We add $n(k+1)$ vertices $y_1^1, \dots, y_{k+1}^1, y_1^2, \dots, y_{k+1}^n$ to G . For every $i \in \llbracket n \rrbracket$, we add positive edges such that the vertices $x_i, y_1^i, \dots, y_{k+1}^i$ induce a positive clique. Let (G', σ') be this new signed graph. We claim that $((G, \sigma), k)$ is a positive instance of EDGE DELETION SIGNED- H_b^1 -COLORING if and only if $((G', \sigma'), k)$ is a positive instance of EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING.

If there exists some $S \subseteq E(G)$ of size at most k such that $(G, \sigma) - S$ is balanced, then $(G', \sigma') - S$ is also balanced. Indeed every cycle of $(G', \sigma') - S$ is either a cycle of (G, σ) , in which case it is balanced by definition of S , or a positive triangle of an added clique, which is also balanced.

If there exists some $S \subseteq E(G)$ of size at most k such that $(G', \sigma') - S \rightarrow_s H_{r,b}^{2-}$, then $(G', \sigma') - S \rightarrow_s H_b^1$. Indeed every vertex of (G', σ') belongs to a positive clique of order $k+2$. This clique is connected in $(G', \sigma') - S$ and contains at least one balanced cycle, hence every vertex of $(G', \sigma') - S$ maps to the positive loop of $H_{r,b}^{2-}$. Hence $((G, \sigma), k)$ is a positive instance of EDGE DELETION SIGNED- H_b^1 -COLORING.

This proves that EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING is NP-complete.

For the parameterized complexity of the two problems EDGE DELETION SIGNED- $H_{-, -}^{2b}$ -COLORING and EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING, we can mimic the proof of the vertex deletion versions to obtain FPT algorithms. We do not describe in length these algorithms as the ideas are exactly the same as in Section 3.4. Nonetheless let us describe some changes.

In order to reduce to the disjoint version, we need to add the edges one by one (instead of vertices) and guess (*i.e.* try all possibilities) for the intersection. As the homomorphism is a function of the vertices and not the edges, we guess the homomorphism on the vertices incident with the removed edges in the solution of size $k+1$. There are at most $2(k+1)$ of them, hence it is still possible to do this in FPT time. After that, it suffices to find cuts instead of separators using the same techniques as for the vertex version. For EDGE DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING, we can use an analogue of Theorem 3.33 (see [53]) for cuts instead of separators to have similar reductions. \square

3.6 Conclusion and perspectives

We have introduced VERTEX DELETION H -COLORING, EDGE DELETION H -COLORING and LIMITED SWITCHINGS (H, π) -COLORING and characterized their complexity for some small H (resp. (H, π)). The full complexity landscape still needs to be determined. We have fully classified the classic complexity of VERTEX DELETION H -COLORING problems. It remains to do the same for EDGE DELETION H -COLORING and LIMITED SWITCHINGS (H, π) -COLORING.

We proved that both VERTEX DELETION H -COLORING and EDGE DELETION H -COLORING are FPT when H has order at most 2. However, if H has order 3, for example if H is a monochromatic triangle, we obtain 3-COLORING, which is not in XP. LIMITED SWITCHINGS (H, π) -COLORING seems particularly interesting, since we obtained an FPT/W[1]-hard dichotomy when (H, π) has order at most 2 (in which case the problem is always in XP). But again for some H of order 3, LIMITED SWITCHINGS (H, π) -COLORING is not in XP. It would be very interesting to obtain FPT/W[1]/XP trichotomies for VERTEX DELETION H -COLORING, EDGE DELETION H -COLORING and LIMITED SWITCHINGS (H, π) -COLORING, at least for some interesting classes of targets H (resp. (H, π)).

We also introduced VERTEX DELETION SIGNED- (H, π) -COLORING and EDGE DELETION SIGNED- (H, π) -COLORING and completely characterized their classic complexity and their parameterized complexity.

One may also study restricted classes of inputs for each of the five problems, such as planar graphs (studied in the context of homomorphisms of signed graphs in [59, 124]). For example, do the W[1]-hard cases of LIMITED SWITCHINGS (H, π) -COLORING become FPT (or even polynomial) when the input is planar?

Another possibility is to allow the removal of vertices and edges in the same problem. A way to define this is to have a bound on the total number of vertices and edges allowed to be removed. One could also decide to have a bound for the vertices and one for the edges. It is very likely that this kind of problem would behave similarly as when only vertices or edges are allowed to be removed.

Chapter 4

Coloring signed graphs with small cyclomatic number

In this chapter, we study the behavior of the chromatic number (and other coloring parameters) of signed graphs on some “simple” classes of graphs.

Up to now, we only considered two types of colorings for signed graphs, coloring of signed graphs and sign-preserving colorings of signed graphs, by studying the two corresponding types of homomorphisms. These two types of colorings are characterized by the two corresponding chromatic numbers χ_s and χ_s^p . Let us introduce two other colorings of signed graphs.

For undirected graphs, a simple way to generalize coloring is to add lists of available colors on each vertex. It is possible to define a similar generalization for signed graphs.

Definition 4.1 (*L-coloring*). A *k-list-assignment* of a signed graph (G, σ) is a function L which assigns to each vertex of G a set of k colors. Recall that when dealing with homomorphisms, we view the vertices of a target graph as colors. Given a *k-list-assignment* L of (G, σ) , an *L-coloring* of (G, σ) is a coloring c of (G, σ) such that for every vertex $v \in V(G)$, we have $c(v) \in L(v)$.

If c is an *L-coloring* of (G, σ) , then we can view c as a homomorphism from (G, σ) to *some* signed graph (H_c, π_c) such that $V(H_c) = \bigcup_{v \in V(G)} L(v)$ and for each vertex $v \in V(G)$, $c(v) \in L(v)$ holds.

The notion of *sign-preserving L-coloring* is defined similarly, requiring that the coloring c is a sign-preserving coloring.

From the notion of list coloring, we can define the choice number.

Definition 4.2 (*Choice number*). The *choice number* of a signed graph (G, σ) , denoted $\text{ch}_s(G, \sigma)$, is the smallest k for which (G, σ) admits an *L-coloring* for any *k-list-assignment* L . Similarly, the *sign-preserving choice number* of a signed graph (G, σ) , denoted $\text{ch}_s^p(G, \sigma)$, is the smallest k for which (G, σ) admits a sign-preserving *L-coloring* for any *k-list-assignment* L .

It is well known that the chromatic number of an undirected graph equals the maximum chromatic number of its connected components. That is why the chromatic number of undirected graphs is generally studied on connected graphs.

This is not true for signed graphs. For example the chromatic number of the signed graph composed of a positive triangle and a negative triangle is 4 (and not 3). This also

holds for the three other parameters: sign-preserving chromatic number, choice number and sign-preserving choice number. This is due to the fact that coloring signed graphs cannot be done locally without considering the rest of the signed graph. For example, when computing the sign-preserving chromatic number of a signed graph, if one edge e has endpoints colored i and j somewhere in the signed graph, then every edge of the signed graph with endpoints colored i and j , no matter how far they are from e , has to have the same sign as e .

For undirected graphs, we usually try to bound the chromatic number by some other parameters. This is for example the case for Brooks' Theorem (see Theorem 1.37) which bounds the chromatic number with the maximum degree of the graph. This is also possible for signed graphs, Das, Nandi, Paul, Sen [54] showed that any signed graph (G, σ) with $\Delta(G) \geq 3$ verifies $\chi_s(G, \sigma) \leq (\Delta(G) - 1)^2 \cdot 2^{\Delta(G)-1} + 2$. Unfortunately, their bound is exponential in $\Delta(G)$, in particular for subcubic graphs, this bound on χ_s is already 18. We are looking for other parameters which can bound the chromatic number of signed graphs.

First, let us see whether there are signed graphs where each connected component is "simple" (*i.e.* trees, cycles, ...) for which one of our coloring parameters is unbounded. The easiest class to consider is the class of signed graphs where each connected component contains a single edge. On this class, the sign-preserving choice number is unbounded.

Theorem 4.3. *For every integer $k \geq 0$, there exists a signed graph (G, σ) such that each connected component of G is an isolated edge and $\text{ch}_s^p(G, \sigma) \geq k$.*

Proof. By a result of Erdős and Rado [65], for every integer k , there exists an integer ℓ_0 such that for every $\ell > \ell_0$ and every function $f : \{1, \dots, \ell\}^2 \rightarrow \{-, 0, +\}$, there exists a subset X of $\{1, \dots, \ell\}$ of cardinality at least k for which $f(X^2)$ is a singleton.

Let k and ℓ be two integers such that ℓ is sufficiently large to apply the aforementioned result. Set $n = \binom{\ell}{k}$ and let (G, λ) be the signed graph composed of $2n$ isolated edges, n of which are positive while the others are negative. The set of positive (resp. negative) edges of (G, λ) is in one-to-one correspondence with the set $\mathcal{P}_k(\{1, \dots, \ell\})$, the set of subsets of size k of $\{1, \dots, \ell\}$. For each such subset X , we denote by e_X^+ (resp. e_X^-) the positive (resp. negative) edge associated with X . We construct the list assignment L by assigning to the two endpoints of the edge e_X^+ (resp. e_X^-) the list X .

Suppose now that (G, σ) admits an L -coloring φ . For each pair of colors (a, b) , either all edges with an endpoint colored a and one endpoint colored b are positive, or they are all-negative, or there is no such edge. Construct the mapping $f_\varphi : \{1, \dots, \ell\}^2 \rightarrow \{-, 0, +\}$ as follows: if there is a positive (resp. negative) edge whose endpoints are colored a and b , then $f_\varphi(a, b) = +$ (resp. $f_\varphi(a, b) = -$); if no such edge exists, then $f_\varphi(a, b) = 0$.

By [65], there exists a set X of size k such that $f_\varphi(X^2)$ is a singleton. Surely, $f_\varphi(X^2) \neq \{0\}$ since the endpoints of the edge e_X^+ are colored. The previous remark even implies $f_\varphi(X^2) = \{+\}$ since e_X^+ is positive. By considering the edge e_X^- , we similarly get $f_\varphi(X^2) = \{-\}$, a contradiction. \square

On this class of graphs, the chromatic number and the choice number are both upper bounded by 2, while the sign-preserving chromatic number is upper bounded by 3. If every connected component is a tree then the chromatic number and the choice number are both upper bounded by 2 as every forest can be switched to be all-positive, reducing the problem to one for undirected graphs. The sign-preserving chromatic number is upper

bounded by 4 for forests as every forest admits a sign-preserving homomorphism to the 4-cycle u, v, w, z for which uv and wz are positive while vw and zu are negative.

Hence, in order to find other interesting classes, we need to have connected components with cycles. The choice number of signed graphs is also unbounded for a rather simple class of signed graphs, namely the class of signed graphs all of whose connected components are unbalanced even cycles.

Theorem 4.4. *For every two integers $k, g \geq 0$, there exists a signed graph (G, σ) , with $g(G) \geq g$ and $\text{ch}_s(G, \sigma) \geq k$, such that each connected component of G is an unbalanced even cycle.*

Proof. The proof of this theorem is similar to the proof of Theorem 4.3. For each set X in $\mathcal{P}_k(\ell)$, we create an unbalanced even cycle of length at least g . The list assignment L associates with each vertex of this cycle the list X .

Suppose now that (G, σ) admits an L -coloring φ , after (possibly) switching the graph. After these switchings, each unbalanced even cycle contains at least one positive edge and one negative edge. By arbitrarily choosing one of the positive (resp. negative) edges of the cycle associated with X to be e_X^+ (resp. e_X^-), we can follow the same arguments as in the proof of Theorem 4.3 and obtain a contradiction. \square

A similar result can be obtained by replacing each unbalanced even cycle by two odd cycles: one balanced (*i.e.* with a positive edge) and one unbalanced (*i.e.* with a negative edge).

Theorem 4.5. *For every two integers $k, g \geq 0$, there exists a signed graph (G, σ) , with $g(G) \geq g$ and $\text{ch}_s(G, \sigma) \geq k$, such that each connected component of G is an odd cycle.*

In the previous construction, we used many cycles to increase the choice number of the signed graph. Intuitively, cycles are the key ingredient to increase the chromatic number of a signed graph. It is then natural to consider the cyclomatic number of the underlying graph, defined as follows.

Definition 4.6 (Cyclomatic number). The *cyclomatic number* of a graph G , denoted by $\nu(G)$, is the minimum number of edges such that the graph obtained from G by deleting these edges is a forest.

It is a basic result that for the cyclomatic number of G , the following formula holds: $\nu(G) = |E(G)| - |V(G)| + c(G)$, where $c(G)$ is the number of connected components of G . The parameter corresponds to the dimension of the cycle space of G : *i.e.* one can interpret it as the minimum number of subgraphs of G having only vertices with even degree required to generate all cycles of the graphs by making sums of these subgraphs. In this context, summing two subgraphs H_1 and H_2 creates the subgraph of G with vertex set $V(G)$ and edge set $E(H_1) \Delta E(H_2)$ (where Δ is the symmetric difference operator).

The cyclomatic number can be found under different names in literature: *nullity*, *circuit rank*, *excess*, or *Betti number*. We denote by \mathcal{B}_i^s , the class of signed graphs having an underlying graph with cyclomatic number at most i . For any signed graph parameter $\lambda \in \{\chi_s, \chi_s^p, \text{ch}_s, \text{ch}_s^p\}$, we denote by $\lambda(\mathcal{B}_i^s)$, the number $\max_{(G, \sigma) \in \mathcal{B}_i^s} \lambda(G, \sigma)$.

The next result shows that it is possible to have high chromatic number, and thus high choice number, with only a few cycles.

Theorem 4.7. *For every integer n , there exists a signed graph (G_n, σ_n) such that*

$$\nu(G_n) = \Theta(\chi_s(G_n, \sigma_n) \cdot \log(\chi_s(G_n, \sigma_n))).$$

Proof. In [80], Füredi, Horák, Pareek and Zhu show that for every q , there exists a signed graph (G_q, σ_q) of order q such that $\chi_s^p(G_q, \sigma_q) = q$ and $|E(G_q)| = \Theta(q \log q)$ (they use the terminology of edge-colored graphs in their article). By construction of these graphs, if u and v are two vertices of G_q , then either uv is an edge or there exists a path of length two uwv such that $\sigma_q(uw) = -\sigma_q(wv)$. Indeed, this is exactly the reason why $\chi_s^p(G_q, \sigma_q) = q$.

We shall now construct the signed graph (G'_n, σ'_n) by adding to (G_{n-1}, σ_{n-1}) a universal vertex all of whose adjacent edges are positive. Hence, if u and v are two vertices of G'_n , then either uv is an edge or there exists an unbalanced cycle of length four containing both u and v . By a result of [154], the signed graph (G'_n, σ'_n) is a signed clique (*i.e.* $\chi_s(G'_n, \sigma'_n) = n$). As (G'_n, σ'_n) has n vertices and $\Theta(n \log n)$ edges, $\nu(G_n) = \Theta(n \log n)$, as required. \square

The aim of this chapter is to provide bounds on the chromatic number, the choice number, and the sign-preserving chromatic number of a signed graph as a function of its cyclomatic number.

Using ideas from [56], we provide an upper bound on the sign-preserving chromatic number of a signed graph in the form of an affine function of its cyclomatic number.

Theorem 4.8. *For every signed graph (G, σ) , $\chi_s^p(G, \sigma) \leq 4 + \nu(G)$.*

A similar result can be obtained for the chromatic number of a signed graph.

Theorem 4.9. *For every signed graph (G, σ) satisfying either $\nu(G) \neq 1$, or $\nu(G) = 1$ and where (G, σ) does not contain an even unbalanced cycle, $\chi_s(G, \sigma) \leq 2 + \nu(G)$.*

Finally, for the choice number, we obtain the following theorem.

Theorem 4.10. *For every signed graph (G, σ) , $\text{ch}_s(G, \sigma) \leq 3 + \nu(G)$.*

Note that in the previous theorem, one could try to replace the $3 + \nu(G)$ by $2 + \nu(G)$ by adding the extra condition that $\nu(G) \geq 2$. However, the proof of such a result would be significantly more complicated and technical. Nonetheless, we prove a better bound when $\nu(G) \leq 2$.

Theorem 4.11. *For every signed graph (G, σ) with $\nu(G) \leq 2$, $\text{ch}_s(G, \sigma) \leq 4$.*

We then determine the exact value of our parameters for some classes of graphs with bounded cyclomatic number.

Theorem 4.12.

1. $\chi_s^p(\mathcal{B}_0^s) = 4$,
2. $\chi_s(\mathcal{B}_0^s) = \text{ch}_s(\mathcal{B}_0^s) = 2$,
3. $\chi_s^p(\mathcal{B}_1^s) = 5$ and $\chi_s^p(\mathcal{B}_2^s) = 6$,
4. if $i = 1$ or 2 , then $\chi_s(\mathcal{B}_i^s) = \text{ch}_s(\mathcal{B}_i^s) = 4$,
5. $\chi_s(\mathcal{B}_3^s) = 5$, and $\chi_s(\mathcal{B}_4^s) = 6$.

It is well known that determining the chromatic number of a graph is NP-complete when the number of colors is at least 3. For signed graphs, we can define the following analogous coloring problem.

COLORING-SIGNED-GRAPHS	Parameter: $\nu(G)$
Input: A signed graph (G, σ) and an integer k .	
Question: Is the number $\chi_s(G, \sigma)$ at most k ?	

Note that the problem COLORING-SIGNED-GRAPHS is NP-complete as shown in [154]. In order to get a better grasp of the complexity of the problem, one might want to study a parameterized variant of the problem. Here, we show that COLORING-SIGNED-GRAPHS is FPT when parameterized by the cyclomatic number of the signed graph.

Theorem 4.13. *The problem COLORING-SIGNED-GRAPHS, on input (G, σ) and k , can be solved in time $2^{O(\nu(G)^2)} \cdot |V(G)|$.*

Note that the previous theorem also implies that the problem of determining the chromatic number of an undirected graph is FPT when parameterized by the cyclomatic number of the graph. Indeed, it is easy to see that $\chi(G) = \chi_s(G, +)$ where $+$ is the signature of G which assigns to each edge of G the positive sign.

This work is joint work with Jan Bok, Nikola Jedličková, Jonathan Narboni and Éric Sopena. This work is partially supported by the ANR project HOSIGRA (ANR-17-CE40-0022) and the IFCAM project “Applications of graph homomorphisms” (MA/IFCAM/18/39).

Section 4.1 focuses on the proofs of the general upper bound theorems: Theorem 4.8, Theorem 4.9 and Theorem 4.10. In Section 4.2, we prove our results involving small cyclomatic numbers, namely Theorem 4.11 and Theorem 4.12. Finally, in Section 4.3, we present our FPT algorithm and prove Theorem 4.13.

Contents

4.1 Proofs of Theorems 4.8, 4.9 and 4.10	124
4.2 Proof of Theorem 4.11 and 4.12	130
4.3 Proof of Theorem 4.13	133
4.4 Perspectives	135

4.1 Proofs of Theorems 4.8, 4.9 and 4.10

This section is devoted to the proofs of each of the three theorems which provide a linear upper bound on some chromatic number of the form $cst + \nu(G)$ where G is the underlying graph of our input signed graph. These proofs are inspired by the proof of a similar type of result on oriented graphs (see [56]). Let us start with the sign-preserving chromatic number.

Proof of Theorem 4.8. A good signed graph (H, π) is a signed graph in which every vertex $v \in V(H)$ is incident with at least one positive edge and at least one negative edge. Let us prove the following stronger statement: every signed graph (G, σ) admits a sign-preserving homomorphism to a good signed graph of order $\nu(G) + 4$. The proof is done by induction

on $(\nu(G), |E(G)| + |V(G)|)$. The case $\nu(G) = 0$ is proven in Theorem 4.12.1. Now suppose $\nu(G) > 0$.

In each of the following cases, we proceed by induction and we always suppose that none of the previous cases applies.

1. *The graph (G, σ) contains a vertex u of degree 1.*

Let v be the neighbor of u . By induction, there exists a sign-preserving homomorphism φ from $(G, \sigma) - u$ to a good signed graph (H, π) of order $\nu(G) + 4$. By definition of a good signed graph, $\varphi(v)$ is incident to an edge $\varphi(v)w$, in (H, π) , with sign $\sigma(uv)$. We can extend φ to (G, σ) by setting $\varphi(u) = w$.

2. *The graph (G, σ) contains a vertex u of degree at least 3 such that $G - u$ has the same number of connected components as G .*

Let x_1, \dots, x_p be the neighbors of u such that ux_i is a positive edge and let y_1, \dots, y_q be the neighbors of u such that uy_i is a negative edge. Without loss of generality, we can suppose $p \geq q$. Note that $\nu(G - u) = \nu(G) - d(u) + 1$, hence $(G, \sigma) - u$ admits a sign-preserving homomorphism φ to a good signed graph (H, π) of order $\nu(G) + 5 - d(u)$. Note that $1 + q \leq 1 + \lfloor \frac{d(u)}{2} \rfloor \leq d(u) - 1$ since $d(u) \geq 3$. To construct our sign-preserving homomorphism φ' , it is sufficient to assign to each vertex of (G, σ) its image by φ , except for u, y_1, \dots, y_q for which we use $1 + q$ vertices among the $d(u) - 1$ available new vertices.

It is then easy to complete the image of $\varphi'(G, \sigma)$ into a good signed graph since (H, π) is good. Indeed, each $\varphi(y_i)$ is incident with a positive edge $\varphi(y_i)x$ where x is a vertex of H . Hence, we can have the same property for $\varphi'(y_i)$ by adding the positive edge $\varphi'(y_i)x$ to $\varphi'(G, \sigma)$. Therefore, only u may not verify the property when $q = 0$ but in that case we have an additional new vertex to use in order to make $\varphi'(G, \sigma)$ good.

3. *The graph (G, σ) contains a cycle C , all whose vertices, except at most one, have degree 2.*

We consider two subcases:

- (a) *The cycle C has a vertex u of degree 2 with v_1 and v_2 being the neighbors of u and $\sigma(uv_1) = \sigma(uv_2)$.*

In this case, let (G', σ') be the signed graph obtained from (G, σ) by removing u . By induction, (G', σ') admits a sign-preserving homomorphism φ to a good signed graph of order $\nu(G') + 4 = \nu(G) + 3$. By adding a new vertex x to $\varphi(G', \sigma')$, we can extend φ to (G, σ) by setting $\varphi(u) = x$. We can then complete $\varphi(G, \sigma)$ into a good signed graph since u has at most two neighbors.

- (b) *C contains an alternating path $uv_1v_2v_3w$ where each v_i has degree 2.*

We then consider the graph (G', σ') obtained from (G, σ) by deleting the vertex v_2 . The vertex v_2 is not a cut-vertex, so we have $\nu(G') = \nu(G) - 1$ and, by induction, (G', σ') admits a sign-preserving homomorphism φ to a good signed graph of order $\nu(G) + 3$.

- If $\varphi(v_1) \neq \varphi(v_3)$, then we add a new vertex x to $\varphi(G', \sigma')$ connected to $\varphi(v_1)$ and $\varphi(v_3)$ with $\sigma(\varphi(v_1)x) = \sigma(v_1v_2)$ and $\sigma(\varphi(v_3)x) = \sigma(v_2v_3)$. We can then extend φ to (G, σ) by setting $\varphi(v_2) = x$. Since $\sigma(v_1v_2) \neq \sigma(v_2v_3)$, the new target graph is a good signed graph and we are done.

- If $\varphi(v_1) = \varphi(v_3)$ and $\sigma(uv_1) = \sigma(\varphi(u)\varphi(w))$, then we add a new vertex x to $\varphi(G', \sigma')$ connected to $\varphi(w)$ and $\varphi(v_1)$ with $\sigma(\varphi(w)x) = \sigma(v_1v_2)$ and $\sigma(\varphi(v_1)x) = \sigma(v_2v_3)$. We then let $\varphi(v_1) = \varphi(w)$ and extend φ to (G, σ) to (G, σ) by setting $\varphi(v_2) = x$, which is possible since $\sigma(uv_1) = \sigma(\varphi(u)\varphi(w))$ and $\sigma(v_3w) = \sigma(\varphi(v_3)\varphi(w))$. Since $\sigma(v_1v_2) \neq \sigma(v_2v_3)$, the new target graph is a good signed graph, and we are done.
- If $\varphi(v_1) = \varphi(v_3)$ and $\sigma(uv_2) = \sigma(\varphi(u)\varphi(w))$, then we add a new vertex x to $\varphi(G', \sigma')$ connected to $\varphi(u)$ and $\varphi(v_1)$ with $\sigma(\varphi(u)x) = \sigma(uv_1)$ and $\sigma(\varphi(v_1)x) = \sigma(v_1v_2)$. We then let $\varphi(v_3) = \varphi(u)$ and extend φ to (G, σ) to (G, σ) by setting $\varphi(v_2) = x$, which is possible since $\sigma(uv_1) = \sigma(\varphi(u)\varphi(v_1))$ and $\sigma(v_3w) = \sigma(\varphi(u)\varphi(w))$. Once again, since $\sigma(v_1v_2) \neq \sigma(v_2v_3)$, the new target graph is a good signed graph.

Note that if C is not an alternating cycle, then subcase (3a) applies and otherwise, subcase (3b) applies.

Consider now a signed graph (G, σ) with $\nu(G) > 0$. By (1), we can suppose that (G, σ) has minimum degree 2. Let G^* be a 2-connected component which is adjacent to at most one other 2-connected component. By the definition of G^* , there is only one vertex $w \in V(G^*)$ which connects G^* to other vertices in $G - G^*$. If G^* is a cycle then case (3) applies. Otherwise, G^* contains a vertex $u \neq w$ with $d_{G^*}(u) \geq 3$. Note that $G - u$ and G have the same number of connected components since $u \neq w$ and G^* is 2-connected. Hence, case (2) applies which concludes the proof. \square

The following proof follows the same methodology, the difficulty in this one lies on the fact that the general formula is not always true. The signed graph UC_4 is one of these exceptions.

Proof of Theorem 4.9. The proof is once again done by induction on $(\nu(G), |E(G)| + |V(G)|)$. Let (G, σ) be a signed graph. We will need the following notion. The *pruned graph* of G , denoted $P(G)$, is the graph obtained from G by repeatedly deleting isolated vertices and vertices of degree 1. Note that $\nu(P(G)) = \nu(G)$. We define $P(G, \sigma)$ in a similar way.

In each of the following cases, we proceed by induction and we always suppose that none of the previous cases applies.

1. *The graph (G, σ) contains a vertex u of degree 1.*

We have $\nu(G - u) = \nu(G)$ and, by induction, there exists a homomorphism φ from $(G, \sigma) - u$ to a signed graph of order $\nu(G) + 2$. Let v be the neighbor of u and w be a neighbor of v different from u . Up to switching u , we can assume $\sigma(uv) = \sigma(vw)$ and extend φ to u with $\varphi(u) = \varphi(w)$.

2. *The signed graph (G, σ) has $\nu(G) \leq 2$.*

If $\nu(G) = 0$ then G is a forest and $\chi_s(G, \sigma) = 2$. If $\nu(G) = 1$ and (G, σ) does not contain an even unbalanced cycle then (G, σ) is a cycle (by the previous point) which is not even unbalanced and $\chi_s(G, \sigma) \leq 3$ as such cycles can be switched to be either all-positive or all-negative. Finally, the case $\nu(G) = 2$ is implied by Theorem 4.11 (the proof of which is independent from the proof of this theorem).

3. *The graph (G, σ) contains a cut-vertex u .*

Let v_1 and v_2 be two neighbors of u that are in different connected components in $G - u$. Up to switching v_1 , we can assume $\sigma(uv_1) = \sigma(uv_2)$. We then consider the graph (H, π) obtained from (G, σ) by identifying v_1 and v_2 . The graph H has less vertices than G and $\nu(H) = \nu(G)$ and so, by induction, there exists a homomorphism φ from (H, π) to a signed graph of order $\nu(G) + 2$. By composing the homomorphism from (G, σ) to (H, π) (obtained from the identification of v_1 and v_2) with φ , we obtain our result.

4. *The graph (G, σ) contains two adjacent vertices u and v of degree 2.*

Let (G', σ') be the graph obtained from (G, σ) by removing u . W.l.o.g., u' is the neighbor of u different from v and v' is the neighbor of v different from u . Note that $\nu(G') = \nu(G) - 1 \neq 1$, hence we can find, by induction, a homomorphism φ from (G', σ') to a signed graph of order $\nu(G) + 1$. We extend φ by fixing $\varphi(u)$ to be our new vertex. If $\varphi(u') = \varphi(v)$ then, up to switching v , one can change $\varphi(v)$ to a vertex different from $\varphi(u)$, $\varphi(v')$ and $\varphi(u')$ since $\nu(G) + 2 \geq 5$. If $\varphi(u') \neq \varphi(v)$ then φ is a homomorphism.

5. *The graph (G, σ) verifies $\nu(G) \geq 3$ and contains a vertex u of degree 2, which is either not part of a UC_4 , or part of a triangle.*

Assume that there exists such a vertex u of degree 2, let x and y be the neighbors of u . We now distinguish two cases.

If x and y are adjacent, then $G - u$ is connected (otherwise (3) would apply), and thus $\nu(G - u) = \nu(G) - 2 + 1 = \nu(G) - 1$. Moreover, as $\nu(G) \geq 3$, we have $\nu(G - u) \geq 2$. So, by induction, there exists a homomorphism from $(G, \sigma) - u$ to a signed graph of order at most $\nu(G) + 1$ colors and we can extend φ by choosing $\varphi(u)$ to be our new vertex.

Otherwise, up to switching x , we have $\sigma(ux) = \sigma(uy)$. Moreover, for any vertex $z \neq u$ in $N(x) \cap N(y)$, we have $\sigma(zx) = \sigma(zy)$. We now consider the graph (H, π) which is the image of (G, σ) by the homomorphism identifying x and y (this is well defined due to the previous remarks). We have $\nu(H) \leq \nu(G) - 2 + 2 = \nu(G)$ and thus, by induction, there exists a homomorphism from (H, π) to a signed graph of order at most $\nu(G) + 2$. Composing the first homomorphism with φ yields the result.

6. *The graph (G, σ) contains a vertex u of degree at least 3 such that $P((G, \sigma) - u)$ is not an even unbalanced cycle.*

Since u is not a cut-vertex, $\nu(G - u) = \nu(G) - d(u) + 1$. If $P((G, \sigma) - u)$ is not an even unbalanced cycle, then by induction, there exists a homomorphism φ from $(G, \sigma) - u$ to a signed graph of order at most $\nu(G - u) + 2 = \nu(G) - d(u) + 3$. Let x_1, \dots, x_p (resp. y_1, \dots, y_q) be the neighbors of u connected to u with positive edges (resp. negative edges). W.l.o.g., we can assume $q \leq p$. Note that $1 + q \leq \left\lfloor \frac{d(u)}{2} \right\rfloor - 1$ as $d(u) \geq 3$. To construct our homomorphism φ' , it is sufficient to assign to each vertex of (G, σ) its image by φ , except for u, y_1, \dots, y_q for which we use $1 + q$ vertices among the $d(u) - 1$ available new vertices.

7. *The graph (G, σ) verifies $\nu(G) \geq 3$ and there exists a UC_4 in (G, σ) containing two vertices of degree 2.*

Let $\{u, v, w, z\}$ be the vertices of this UC_4 such that u and v have degree 2. If u and v are adjacent then (4) applies. If u and v are not adjacent and w and z are

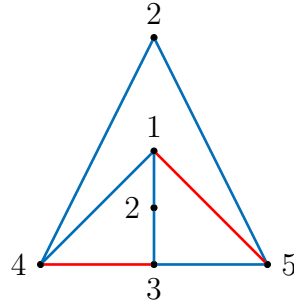


Figure 4.1: The graph involved in the proof of Theorem 4.9

adjacent then (5) applies.

If u and v are not adjacent and w and z are not adjacent then let (G', σ') be the graph obtained from (G, σ) by removing u and v and adding the edge wz . Note that $\nu(G') = \nu(G) - 1 \neq 1$, hence we can find a coloring c of (G', σ') by induction. We want to extend c to u and v . Note that $c(v) \neq c(w)$. Let c_0 be a color different from $c(w)$ and $c(z)$. W.l.o.g suppose that u has both its incident edges of the same sign and that both edges incident with v have opposite signs when trying to extend the coloring c . Note that all edges which have endpoints colored c_0 and $c(v)$ have the same signs, and the same property holds for all the edges which have endpoints colored c_0 and $c(w)$. If the edges which have endpoints colored c_0 and $c(w)$ have the same sign as the edges which have endpoints colored c_0 and $c(v)$, then, up to switching u , we can color u with c_0 and assign our new color to v . If the edges which have endpoints colored c_0 and $c(w)$ do not have the same sign as the edges which have endpoints colored c_0 and $c(v)$, then up to switching v , we can color v with c_0 and assign our new color to u . In both cases, we extend c to a signed graph equivalent to (G, σ) .

8. *The graph (G, σ) contains a vertex s of degree at least 5.*

Since $P((G, \sigma) - s)$ is an unbalanced even cycle by (6), $(G, \sigma) - s$ admits a homomorphism φ to a signed graph of order 4 and we still have $d(s) - 2$ new vertices to extend the coloring to (G, σ) . Let x_1, \dots, x_p (resp. y_1, \dots, y_q) be the neighbors of s connected to s with positive edges (resp. negative edges). Without loss of generality, we can assume $q \leq p$. As $d(s) \geq 5$, we have $d(s) - \lfloor \frac{d(s)}{2} \rfloor - 2 \geq 1$. To construct our homomorphism φ' from (G, σ) , it is sufficient to assign to each vertex of (G, σ) its image by φ , except for u, y_1, \dots, y_q for which we use $1 + q$ vertices among the $d(u) - 1$ available new vertices.

9. *The graph (G, σ) contains a vertex s of degree 4.*

By (6) and (8), (G, σ) has maximum degree 4 and $(G, \sigma) - s$ contains an unbalanced even cycle UC_{2p} for some integer p . Recall that G is 2-connected by (3). If $\nu(G) \neq 4$ then $\nu(G - s) = \nu(G) - 3 \neq 1$ and (6) applies. If G has a vertex x of degree 3, then $\nu(G - x) = 4 - 2 = 2$ and (6) applies. As $\nu(G) = 4$, there are three vertices of degree 4 in G and all the other vertices have degree 2. If $p \geq 3$ then (G, σ) admits a vertex of degree 2 which is not part of a UC_4 and (5) applies. Hence $p = 2$ and this UC_4 has two vertices of degree 2 which contradicts (7).

10. *The graph (G, σ) contains a vertex s of degree 3.*

We denote by n_i the number of vertices of degree i in G . We have:

$$\nu(G) = m - n + 1 = \frac{3 \cdot n_3}{2} + \frac{2 \cdot n_2}{2} - n_3 - n_2 + 1 = \frac{n_3}{2} + 1.$$

Since G is 2-connected by (3), deleting a vertex of degree 3 decreases the cyclomatic number $\nu(G)$ by 2. Moreover, since $P(G - v)$ is an unbalanced even cycle for all vertices v of degree 3 (by 6), $\nu(G - v) = 1$. Hence we have $\nu(G) = 3$ and $n_3 = 4$.

If $(G, \sigma) - s$ contains an unbalanced even cycle UC_{2p} for some integer $p \geq 3$ then (G, σ) admits a vertex of degree 2 which is not part of a UC_4 and (5) applies. Hence for all vertices v of degree at least 3, $P((G, \sigma) - v) = UC_4$.

Since $P((G, \sigma) - s) = UC_4$, (G, σ) contains a UC_4 denoted C on the vertex set $\{u, v, w, z\}$. If C has at least two vertices of degree 2 then (7) applies. Hence C has at least three vertices of degree 3, say u, v and w . Since $d(z) = 2$, we can suppose w.l.o.g. that u is the nearest vertex from s in C .

We first show that $d(u, s) = 1$. If $d(u, s) \geq 3$ then there are two adjacent vertices of degree 2 along this path and (5) or (7) applies. If $d(u, s) = 2$, then there is a path uyx , where y has degree 2, and thus y should be part of a UC_4 (by 5) which is only possible if s is adjacent to a vertex of C ; a contradiction.

So $us \in E(G)$, and we have to distinguish whether u is a neighbor of z or not. If $uz \notin E(G)$, then, as $P((G, \sigma) - u)$ is a UC_4 , v and w have another common neighbor y , and this neighbor can only be s , otherwise u would be a cut-vertex and (3) would apply. So $\{sv, sw\} \subseteq E(G)$, but this contradicts the fact that $P((G, \sigma) - v)$ is a UC_4 .

Hence $uz \in E(G)$, there is one vertex from $\{v, w\}$ which is not a neighbor of u , say $uw \notin E(G)$. As $P(G - v)$ is a UC_4 , then there should be an edge between s and w ; and as $P(G - w)$ is also a UC_4 , there exists a vertex y of degree 2 neighboring s and v . There is a signature corresponding to this graph which has all the edges positive except for $\{sw, uv\}$. This graph can be easily colored with five colors as depicted in Figure 4.1.

There are no other cases. □

Another similar result can be obtained for the choice number of signed graphs. The difficulty here is that we do not have “unused colors” at our disposal by simply applying the induction. Hence we have to carefully manipulate the lists of each vertex to obtain our result.

Proof of Theorem 4.10. The proof is done by induction on $(\nu(G), |E(G)| + |V(G)|)$. Let (G, σ) be a signed graph of order n and let $L : V(G) \rightarrow \mathcal{P}(\mathbb{N})$ be a list assignment on the vertices of G where the lists have size at least $3 + \nu(G)$.

With an L -coloring c of (G, σ) , we associate a signed graph (H_c, π_c) where H_c is a graph on the vertex set $\{c(u), u \in V(G)\}$ such that two colors are adjacent if and only if there exists in G two adjacent vertices colored by these two colors. The signature π_c verifies that, for any two colors c_1 and c_2 , $\pi_c(c_1c_2)$ is the sign of all edges uv of G such that $c(u) = c_1$ and $c(v) = c_2$.

In the following cases, we can apply induction. When considering a particular case, we suppose that none of the previous cases applies. Finally, note that one can make similar arguments by inverting the roles of positives edges and negatives edges.

1. G has a degree 1 vertex.

Let u be a vertex of degree 1 of G and let v be the neighbor of u . By induction, there exists a signed graph (G', σ') equivalent to $(G, \sigma) - u$ which admits an L -coloring c . Let c_u be a color in $L(u)$. By switching u , we can suppose that uv has the same sign as $\pi_c(c_u c(v))$ and extend the coloring to (G, σ) .

2. G has two adjacent vertices u and v of degree 2.

Let u' (resp. v') be the neighbor of u (resp. v) which is not v (resp. u). Choose $c_u \in L(u)$ and, for every $w \in V(G) \setminus \{u\}$, let $L'(w) = L(w) \setminus \{c_u\}$. By induction, there exists a signed graph (G', σ') equivalent to $(G, \sigma) - u$ which admits an L' -coloring c . Color u with c_u .

If $c(u') \neq c(v)$ then c is a coloring of a signed graph equivalent to (G, σ) . Otherwise, let A be the set of colors $\alpha \in L(v) \setminus \{c(u'), c_u\}$ for which $\pi_c(\alpha c(v')) = \sigma(vv')$. If A is non-empty then we can recolor v with a color in A . Otherwise, switch v , and color v with an arbitrary color of $L'(v)$ which is different from $c(u')$.

3. There is a vertex u of degree at least 3 such that $G - u$ has the same number of connected components as G .

Let v_1, \dots, v_k be the k neighbors of u . Let $c_u \in L(u)$, $c_4 \in L(v_4) \setminus \{c_u\}$, $c_5 \in L(v_5) \setminus \{c_u, c_4\}$, \dots , and $c_k \in L(v_k) \setminus \{c_u, c_4, c_5, \dots, c_{k-1}\}$. For every $w \in V(G) \setminus \{u\}$, let $L'(w) = L(w) \setminus \{c_u, c_4, c_5, \dots, c_k\}$. Let α be a color of $L'(v_1) \cup L'(v_2) \cup L'(v_3)$ which appears in the greatest number of sets among $L'(v_1)$, $L'(v_2)$ and $L'(v_3)$. Up to changing the order of v_1, v_2 and v_3 , we can suppose that there exists $\ell \in \{1, 2, 3\}$ such that $\alpha \in L'(v_i)$ for $i \leq \ell$ and $\alpha \notin L'(v_i)$ for $\ell < i \leq 3$. For every $w \in V(G) \setminus \{u\}$, let $L''(w) = L'(w) \setminus \{\alpha\}$. Note that we removed at most $d(u) - 1$ colors from each list from L to L'' .

As $\nu(G - u) = \nu(G) - (d(u) - 1)$, by induction, there exists a signed graph (G', σ') equivalent to $(G, \sigma) - u$ which admits an L'' -coloring c . We extend c to the vertex u , in two steps. First assign the color c_u to u and the color c_i to v_i where $3 < i \leq k$. When doing so, the obtained coloring may have some problems: it is possible for a color β to be assigned to two neighbors v_i and v_j of u such that uv_i is a positive edge and uv_j is a negative edge. In this case, $i, j \in \{1, 2, 3\}$. If this happens then there are two subcases.

- (a) The vertices v_1, v_2 and v_3 all receive the same color.

In this case, $\ell = 3$ and either all uv_i , $i \in \{1, 2, 3\}$ have the same sign or one of these three edges, say uv_j , has a sign different from the other two. In the first case, there is no problem with the coloring, in the second case, one can recolor v_j with α .

- (b) Two of the three vertices, say v_i and v_j , receive the same color.

In this case, $\ell \geq 2$ and one can recolor v_2 with α . This removes the conflict as $2 \in \{i, j\}$.

Take any signed graph (G, σ) . By (1), we can suppose that (G, σ) has minimum degree 2. Let G^* be a 2-connected component which is adjacent to at most one other 2-connected component. By definition of G^* , there is only one vertex $w \in V(G^*)$, which connects G^* to other vertices in $G - G^*$. If G^* is a cycle then case (2) applies. Otherwise, G^* contains a vertex u with $d_{G^*}(u) \geq 3$ different from w . Note that $G - u$ and G have

the same number of connected components since $u \neq w$ and G^* is 2-connected. Hence, case (3) applies. \square

4.2 Proof of Theorem 4.11 and 4.12

This section presents the proofs of results where we compute exact values of our coloring parameters when the cyclomatic number of the graph is small. As mentioned before, Theorem 4.10 could be improved by taking into account the exception of Theorem 4.9. Theorem 4.11 provides a better upper bound when $\nu(G) \leq 2$.

Proof of Theorem 4.11. The proof is done by induction on $(\nu(G), |E(G)| + |V(G)|)$. Let (G, σ) be a signed graph of order n with $\nu(G) \leq 2$ and let $L : V(G) \rightarrow \mathcal{P}(\mathbb{N})$ be a list assignment on the vertices of G where the lists have size at least 4.

With an L -coloring c of (G, σ) , we associate a signed graph (H_c, π_c) as in the proof of Theorem 4.10.

In the following cases, we can apply induction. When considering a particular case, we suppose that none of the previous cases applies. Finally, note that one can make similar arguments by inverting the roles of positive edges and negative edges.

1. G has a degree 1 vertex.

Let u be a vertex of degree 1 of G and let v be the neighbor of u . By induction, there exists a signed graph (G', σ') equivalent to $(G, \sigma) - u$ which admits an L -coloring c . Let c_u be a color in $L(u)$. By switching u , we can suppose that uv has the same sign as $\pi_c(c_u c(v))$ and extend the coloring to (G, σ) .

2. (G, σ) is equivalent to an all-positive signed graph (G, σ') .

The signed graph (G, σ') is all-positive and is thus 3-choosable since $\nu(G) \leq 2$.

From this point onward, we can suppose that G is composed of either exactly one vertex of degree 4 and $n - 1$ vertices of degree 2, or exactly two vertices of degree 3 and $n - 2$ vertices of degree 2, or only vertices of degree 2. As $\nu(G) = \sum_{v \in V(G)} \frac{d(v)-2}{2} + C(G) \leq 2$,

$C(G) \geq 1$ and $\frac{d(v)-2}{2} \geq 0$ for every $v \in V(G)$, if there is a vertex of degree 4 then all other vertices have degree 2, if there is a vertex of degree 3 then there must be another vertex of odd degree (which then has degree 3) and all other vertices must have degree 2.

3. (G, σ) is equivalent to a signed graph (G, σ') which has two edges uv and wz where $d(u) = d(v) = d(w) = d(z) = 2$, $G - u - z$ is a forest and such that beside uv and wz , all other edges of (G, σ) are positive.

Let u' be the neighbor of u which is not v and let z' be the neighbor of z which is not w . Choose $c_u \in L(u)$ and $c_z \in L(z)$ such that $c_u \neq c_z$. Let L' be the list assignment obtained from L by removing the colors c_u and c_z from every set. Since $(G, \sigma') - u - z$ is an all-positive signed forest, it admits an L' -coloring c as ordinary forest are 2-choosable. We extend c to (G, σ') by coloring u by the color c_u and z by the color c_z . Some problems may arise from this choice. If $c(u') = c(v)$ then c is not a coloring of (G, σ') . One can recolor v by a color in $L(v) \setminus \{c(u'), c_u, c_z\}$, in order for the previous case to be avoided. The same holds if $c(z') = c(w)$.

4. (G, σ) is equivalent to a signed graph (G, σ') which has exactly one negative edge uv where $d(u) = d(v) = 2$.

In this case, we can do the same analysis as in case 3 by simply ignoring the vertex z . As $G - u$ is all-positive, with at most one cycle, $(G, \sigma) - u$ is 3-choosable and we color (G, σ) with the same technique.

5. (G, σ) has a vertex x of degree 4.

In this case, G consists of two edge-disjoint cycles with a common vertex x . By switching some vertices of (G, σ) , we can ensure that x is incident with only positive edges and that the obtained signed graph has at most two negative edges. In this case, we can apply case 3.

Let (G, σ') be a signed graph equivalent to (G, σ) with the least number of negative edges. The cases (2) and (4), ensure that (G, σ') has at least two negative edges uv and zw . Since $\nu(G) \leq 2$, (G, σ') has exactly two negative edges. If we cannot apply case (3), then either one of u, v, w or z is not of degree 2 or $\nu(G - u - z) \geq 1$. If $\nu(G - u - z) \geq 1$ then there exists a path of vertices of degree 2 between u and z . By switching the vertices of this path, we can obtain a signed graph with no negative edges and apply case (2). By switching some vertices, since there are at most two vertices of degree at least 3, one can ensure that $d(u) = d(v) = d(w) = d(z) = 2$. In this case, we can apply case (3). \square

We now prove Theorem 4.12. Let us start with the case of forests with Theorem 4.12.1 and Theorem 4.12.2.

Proof of Theorem 4.12.1. Nešetřil and Raspaud in [157, Theorem 1.1] showed that the chromatic number of 2-edge-colored forests is upper bounded by 4. By bijection between 2-edge-colored graphs and signed graphs, we obtain $\chi_s^p(\mathcal{B}_0^s) \leq 4$.

Suppose to the contrary that $\chi_s^p(\mathcal{B}_0^s) \leq 3$. Then every signed forest admits a sign-preserving homomorphism to some (K_3, π) . Let (F, σ) be the signed forest composed of eight disjoint path of length 3. Each of these path will have a unique signature among the eight possible ones. Let (K_3, π) such that (F, σ) admits a sign-preserving homomorphism to (K_3, π) . As the all-positive (resp. all-negative) path of length 3 is in (F, σ) , (K_3, π) cannot be all-negative (resp. positive). Suppose w.l.o.g. that (K_3, π) has two positive edges ab and ac and a negative edge bc . Then the path of length 3, say uvw , with the signature $\sigma(uv) = \sigma(wx) = -$ and $\sigma(vw) = +$ does not admit a homomorphism to (K_3, π) , a contradiction.

Hence $\chi_s^p(\mathcal{B}_0^s) = 4$. \square

Proof of Theorem 4.12.2. First remark that $\chi_s(\mathcal{B}_0^s) > 1$ and $\text{ch}_s(\mathcal{B}_0^s) > 1$ as signed graphs in these classes contain edges. Recall that all signed forests can be switched to have only positive edges. These forests can be treated as ordinary graph and thus their chromatic number is 2 and their choice number is also 2. Hence our result. \square

We now focus on the sign-preserving chromatic number and provide lower bounds when $\nu(G) = 1$ and $\nu(G) = 2$.

Proof of Theorem 4.12.3. First note that $\chi_s^p(\mathcal{B}_1^s) \leq 5$ and $\chi_s^p(\mathcal{B}_2^s) \leq 6$ by Theorem 4.8.

Consider the alternating cycle (C_6, σ) of length 6 (see Figure 4.2(a)). Any possible identification of a pair of vertices of this C_6 involves two antipodal vertices. Once identified, without loss of generality, we obtain the signed graph with five vertices a, b, c, d and

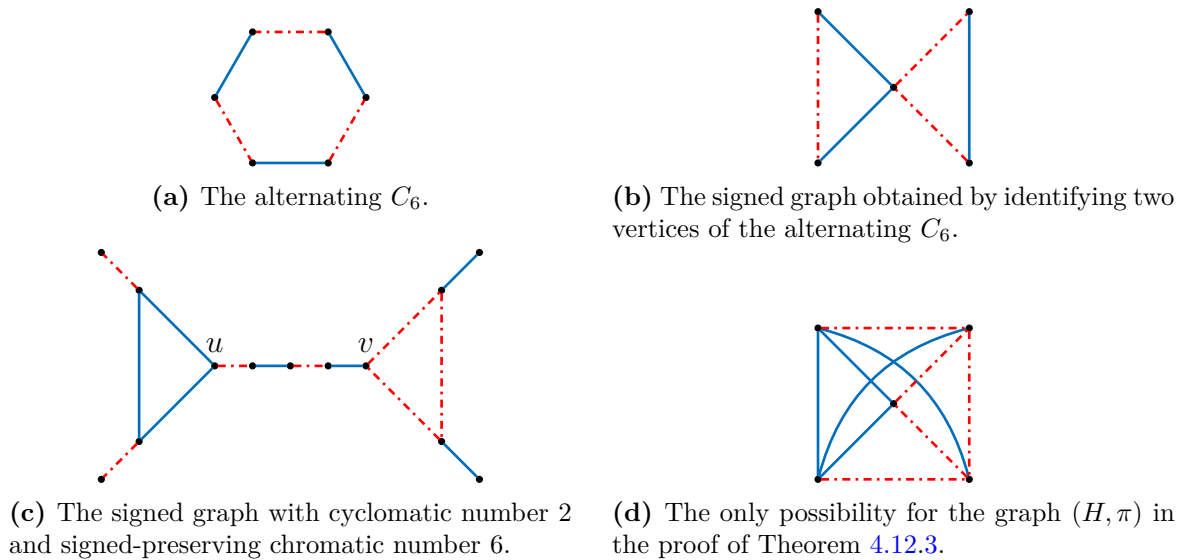


Figure 4.2: The graphs involved in the proof of Theorem 4.12.3.

x where xa , xb and cd are the positive edges and xc , xd and ab are the negative edges (see Figure 4.2(b)). This graph is a sign-preserving clique. Hence $\chi_s^p(C_6, \sigma) = 5$.

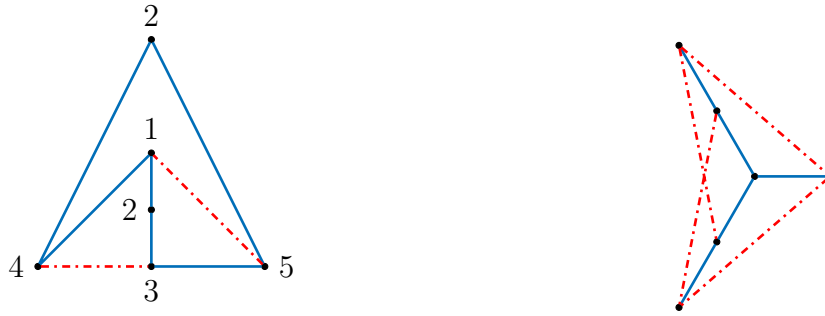
Let (G, σ) be the signed graph represented in Figure 4.2(c). We claim that we have $\chi_s^p(G, \sigma) \geq 6$. By contradiction, suppose $\chi_s^p(G, \sigma) \leq 5$, that is, (G, σ) admits a sign-preserving homomorphism to a complete graph (H, π) on five vertices. The signed graph (H, π) has at least one all-positive triangle and one all-negative triangle. These two are the images of the two triangles of (G, σ) . Note that each vertex of the two triangles of (G, σ) is adjacent to at least one positive and one negative edge, hence it must also be the case for every vertex of (H, π) . Up to some permutation of the vertices, there is only one possibility for (H, π) which is represented on Figure 4.2(d). Note that the image of u lies on the positive triangle while the image of v lies on the negative triangle. In (G, σ) , there is an alternating path of length 4 between the vertices u and v , hence in (H, π) there must be one such path between their images. There is no such path in (H, π) , a contradiction. \square

We continue with the cases where $\nu(G) \leq 2$ for both the chromatic number and the choice number of signed graphs.

Proof of Theorem 4.12.4. Theorem 4.11 gives the upper bound for both parameters (recall that $\chi_s(G, \sigma) \leq ch_s(G, \sigma)$ for every signed graphs (G, σ)). Now, consider the signed graph UC_4 , it has chromatic number and choice number equal to 4, and cyclomatic number equal to 1. Hence $\chi_s(\mathcal{B}_1^s) = ch_s(\mathcal{B}_1^s) = \chi_s(\mathcal{B}_2^s) = ch_s(\mathcal{B}_2^s) = 4$. \square

We finish with the cases where $\nu(G) \in \{3, 4\}$ for the chromatic number of signed graphs.

Proof of Theorem 4.12.5. First note that $\chi_s(\mathcal{B}_3^s) \leq 5$ and $\chi_s(\mathcal{B}_4^s) \leq 6$ by Theorem 4.9. To show the lower bounds, consider the two graphs of Figure 4.3. On the signed graph in Figure 4.3(a), only one identification is possible (even with switching): the identification of the two vertices colored 2. After identification, we obtain a signed clique and thus the chromatic number of this signed graph is 5. The signed graph in Figure 4.3(b) is a signed



(a) A signed graph (G, σ) with $\nu(G) = 3$ and $\chi_s(G, \sigma) = 5$. (b) A signed graph (G, σ) with $\nu(G) = 4$ and $\chi_s(G, \sigma) = 6$.

Figure 4.3: Two signed graphs used in the proof of Theorem 4.12.5.

clique as for every pair of vertices u and v , either u and v are adjacent or they belong to some UC_4 . Hence, the chromatic number of this signed graph is 6. \square

4.3 Proof of Theorem 4.13

We present our algorithm solving COLORING-SIGNED-GRAPHS in FPT time when parameterized by the cyclomatic number of the input graph. Note that computing the sign-preserving chromatic number can be done in a similar fashion by ignoring switching is the proof.

Proof of Theorem 4.13. Let \mathcal{I} be the instance of COLORING-SIGNED-GRAPHS composed of the signed graph (G, σ) of order n and the integer k .

First, note that if $k \geq \nu(G) + 3$, then $\chi_s(G, \sigma) \leq k$ by Theorem 4.9 and we can accept our instance \mathcal{I} . Therefore, in the rest of the proof, we can assume $k = O(\nu(G))$.

Observe that, by definition, if $\chi_s(G, \sigma) \leq k$, then $(G, \sigma) \rightarrow_s (H, \pi)$ for some signed graph (H, π) of order at most k . Note that we can always complete H so that $H = K_k$. There are at most $2^{\binom{k}{2}}$ such signed graphs on k vertices. For each of them, we test whether $(G, \sigma) \rightarrow_s (H, \pi)$ or not. This step is described below and is repeated at most $2^{\binom{k}{2}}$ times.

From now on fix (H, π) to be one of the $2^{\binom{k}{2}}$ candidate targets for the homomorphism. Fix a spanning forest F of G . The forest F can be computed in time $O(n + \nu(G))$ through e.g. a breadth-first search algorithm. Based on F , we partition $V(G)$ into four sets of vertices in the following way.

$$\begin{aligned} A &= \{v \in V(G) : d_F(v) \neq d_G(v)\}, \\ B_1 &= \{v \in V(G) : d_F(v) = d_G(v) = 1\}, \\ B_2 &= \{v \in V(G) : d_F(v) = d_G(v) = 2\}, \text{ and} \\ B_3 &= \{v \in V(G) : d_F(v) = d_G(v) \geq 3\}. \end{aligned}$$

First remark that $(G, \sigma) \rightarrow_s (H, \pi)$ if and only if $(G, \sigma) - B_1 \rightarrow_s (H, \pi)$ when $k \geq 2$. Indeed, we can always, possibly with some switching, map a vertex u of degree 1 to a vertex of H which is not the image of the neighbor of u . Hence by successively removing vertices of degree 1 from G , we can assume that B_1 is empty.

Note that $|A| \leq 2\nu(G)$ as there are at most $2\nu(G)$ vertices incident with one of the $\nu(G)$ edges which do not belong to F . Note that each leaf of F belongs to A since B_1 is empty. Finally, note that in every tree, the number of internal vertices of degree at least 3 is bounded by the number of leaves of this tree. Thus, $|B_3| \leq |A| \leq 2\nu(G)$.

If $(G, \sigma) \rightarrow_s (H, \pi)$, then each of the vertices of $A \cup B_3$ has an image in (H, π) . We test, for each possible assignment of images to the vertices of $A \cup B_3$, whether they lead to a homomorphism from (G, σ) to (H, π) or not. There are $O((2k)^{4\nu(G)})$ possible assignments of such kind to consider. Indeed, H is of size k and each of the $O(4\nu(G))$ vertices of $A \cup B_3$ may be switched in the homomorphism. Let φ be a function which assigns images in $V(H)$ to the vertices of $A \cup B_3$ and possibly switch some vertices of $A \cup B_3$. If we can check whether any of the $O((2k)^{4\nu(G)})$ possible φ functions can be extended to the vertices of B_2 , then we are done. Some of these functions are already not homomorphisms and hence we can ignore them (they can be detected in $O(m) = O(n + \nu(G))$ time).

To check whether φ extends to B_2 , it suffices to check that φ can be extended to every maximal path $P = uw_1w_2 \dots w_tv$, where $u, v \in A \cup B_3$ and $w_i \in B_2$ for each $i \in \{1, \dots, t\}$. Note that there are $O(n)$ such paths. To check whether we can find images for the w_i 's of P , it suffices to check whether there exists a walk in (H, π) starting at $\varphi(u)$ and ending at $\varphi(v)$ with length $t + 1$ and with sign equal to $\sigma(P)$.

Let $\Gamma(H, \pi)$ be the ordinary graph with vertex set $\{+1, -1\} \times \{0, 1\} \times V(H)$ where two vertices $(\varepsilon_1, 0, v_1)$ and $(\varepsilon_2, 1, v_2)$ are adjacent if and only if $v_1v_2 \in E(H)$ and $\varepsilon_1\varepsilon_2 = \pi(v_1v_2)$. Two vertices (ε_1, i, v_1) and (ε_2, i, v_2) for $i \in \{0, 1\}$ are not adjacent. There is a walk of length ℓ and of sign ε between two vertices u and v of H if and only if the distance between $(+, 0, u)$ and (ε, i, v) in $\Gamma(h, \pi)$ is at most ℓ where $i = 0$ if ℓ is even and $i = 1$ otherwise. Hence, it suffices to compute the distances between the vertices of $\Gamma(H, \pi)$ to find the answer and that can be done in time $O(\nu(G)^3)$.

To recapitulate, for each possible target (H, π) , we consider each possible mapping of $A \cup B_3$ to $V(H)$ and we try to extend this to a homomorphism from (G, σ) . To do that it suffices to check for each maximal path of vertices of B_2 whether there is a corresponding walk in (H, π) and this can be done by computing the distances in the graph $\Gamma(H, \pi)$. If we find one such homomorphism, we can accept and otherwise, we reject.

In the end, our algorithm runs in time

$$2^{\binom{\nu(G)}{2}} \times (2\nu(G))^{4\nu(G)} \times O(\nu(G)^3) \times O(n + \nu(G)) = 2^{O(\nu(G)^2)} n.$$

This concludes the proof. □

4.4 Perspectives

We proved upper bounds on some chromatic numbers of signed graphs which are linear in the cyclomatic number of the graph. One interesting question is whether these bounds are asymptotically optimal? We showed that some signed graphs have cyclomatic number of the order of $k \log k$ where k is the chromatic number of the signed graph. Hence there is a gap between this value and our current linear upper bound, in particular we do not know of a family of signed graphs whose chromatic number is linear in its cyclomatic number. We raise the following conjecture.

Conjecture 4.14. *For every signed graph (G, σ) , if $k = \chi_s(G, \sigma)$ (resp. $k = \chi_s^p(G, \sigma)$, resp. $k = \text{ch}_s(G, \sigma)$), then:*

$$k \log(k) = O(\nu(G)).$$

It would also be interesting to determine upper bounds on the chromatic number (or the other variants) of a signed graph in its number of cycles. It is known that the number of cycles of a graph can be exponential in the number of edges (and thus in the cyclomatic number) of the graph, hence we should expect the chromatic number (or the other variants) of a signed graph to be smaller than some function on the order of magnitude of the logarithm of its number of cycles. Theorem 4.4 uses $\binom{\ell}{k}$ cycles in order to have a signed graph with choice number at least k , where $\ell = 3^{3^{O(k)}}$ according to [65]. This number of cycles is way over blown and it would be interesting to know what is the minimum number of cycles that we need to use to obtain this type of result.

Chapter 5

Cartesian product of signed graphs

We are interested in the study of Cartesian products of signed graphs, defined by Germina, Hameed and Zaslavsky in [85]. In their paper, the three authors mainly study the spectral properties of the Cartesian product, *i.e.* the impact of the Cartesian product on the spectre of the adjacency matrices of signed graphs. In this chapter, we present algebraic properties of the Cartesian product and study the chromatic number of some Cartesian products of signed graphs.

Cartesian products of graphs are useful to represent crystalline structures in which some metals can aggregate themselves. The study of the Cartesian product of signed graphs is particularly useful in the context of the Ising model (see Section 2.1.4) used to represent ferromagnetic materials.

The Cartesian product of two ordinary graphs G and H , noted $G \square H$, has been extensively studied. In 1957, Sabidussi [168] showed that $\chi(G \square H) = \max(\chi(G), \chi(H))$ where $\chi(G)$ is the chromatic number of the graph G . Another notable article on the subject by Sabidussi [169] shows that every connected graph G admits a unique prime decomposition, *i.e.* there is a unique way to write a graph G as a Cartesian product of some graphs up to isomorphism of the factors. This result was also independently discovered by Vizing in [183]. Another algebraic property, the cancellation property, which states that if $A \square B = A \square C$, then $B = C$, was proved by Imrich and Klavžar [114] using a technique of Fernández, Leighton and López-Presa [74]. On the complexity side, the main question associated with the Cartesian product is to decompose a graph with the best possible complexity. The complexity of this problem has been improved successively in [72, 195, 70, 10] to finally reach an optimal complexity of $O(m)$ in [115] where m is the number of edges of the graph.

An extended abstract of this chapter is published in the proceedings of the international conference CALDAM 2020 [137]. A full version of the corresponding paper is also published in Discrete Applied Mathematics [138]. We would like to thank the reviewers of our submission to CALDAM 2020 for their comments, especially Reviewer 2 of our submission to CALDAM 2020 for pointing us to the techniques of [116] which improved our algorithm. We would also like to thank the reviewers of the journal version.

This work is partially supported by the ANR project HOSIGRA (ANR-17-CE40-0022) and the IFCAM project “Applications of graph homomorphisms” (MA/IFCAM/18/39).

Our study of the Cartesian product of signed graphs is divided in several sections. In section 5.1, we present some useful results on the Cartesian product of undirected graphs. In section 5.2, we present the definition of the Cartesian product of signed graphs and give

some first properties and easy consequences of the definition. We also prove the prime decomposition theorem for signed graphs and give an algorithm to decompose a Cartesian product of signed graphs into its factors. We study the chromatic number of Cartesian products of signed complete graphs in section 5.3 and products of cycles in section 5.4. Finally, we present some open problematics in Section 5.5.

Contents

5.1	Definitions and preliminary results	138
5.2	Cartesian products of signed graphs	139
5.2.1	Definition	139
5.2.2	Signed grids	141
5.2.3	Prime factor decomposition: existence and unicity	143
5.2.4	Recognising Cartesian products of signed graphs	146
5.3	Chromatic number of Cartesian products of complete signed graphs and upper bounds	149
5.3.1	s -redundant sets	150
5.3.2	Back to Cartesian products of complete signed graphs	151
5.4	Chromatic number of Cartesian products of signed cycles	153
5.4.1	Signed complete graphs of order 4	155
5.4.2	Beginning of the proof of Lemma 5.30	155
5.4.3	Number of turns in P	156
5.4.4	Regions induced by a coloring of (P, π)	158
5.4.5	Crossings and turns	165
5.4.6	End of the proof	167
5.5	Perspectives	167

5.1 Definitions and preliminary results

Let us start with definitions.

Definition 5.1 (Cartesian product). The *Cartesian product* of two ordinary graphs G and H is the graph $G \square H$ whose vertex set is $V(G) \times V(H)$ and where (x, y) and (x', y') are adjacent if and only if $x = x'$ and y is adjacent to y' in H , or $y = y'$ and x is adjacent to x' in G .

A graph can be written as a product of multiple other graphs. We present a tool useful in characterizing a product.

Definition 5.2 (Prime decomposition). A *decomposition* D of a graph G is a multiset $\{G_1, \dots, G_k\}$, $k \geq 1$, such that the G_i 's are graphs having at least two vertices and $G = G_1 \square \dots \square G_k$. The G_i 's are called *factors* of G . A graph G is *prime* if there are no graphs A and B on at least two vertices for which $G = A \square B$. A decomposition is *prime* if all the G_i 's are prime.

We can also compare distinct decompositions of the same graph. A decomposition D' is *finer* than a decomposition $D = \{G_1, \dots, G_k\}$, if for all $i \in \llbracket k \rrbracket$, there is a decomposition $D'_i = \{G'_{i,1}, \dots, G'_{i,p_i}\}$ of G_i such that $D' = \{G'_{1,1}, \dots, G'_{1,p_1}, G'_{2,1}, \dots, G'_{k,p_k}\}$. Note that by definition, every decomposition is finer than itself.

Suppose that G is a graph and $D = \{G_1, \dots, G_k\}$ is a decomposition of G such that $G = G_1 \square \dots \square G_k$. It is useful to add to G some more structure by identifying the product structure of the graph. To the end, we use the concepts of coordinate systems and layers.

Definition 5.3 (Coordinate system). A *coordinate system* for G under the decomposition D is a bijection $\theta : V(G) \rightarrow \prod_{i=1}^k V(G_i)$ verifying that for each vertex v of G , the set of vertices which differ from v by the i th coordinate induces a graph, noted G_i^v and called a G_i -*layer*, which is isomorphic to G_i by the projection on the i th coordinate.

An edge uv of G is a *copy* of an edge ab of G_i if $\theta(u)$ and $\theta(v)$ differ only in their i th coordinate with $u_i = a$ and $v_i = b$. For a vertex u of G and a G_i -layer G_i^v , the projection of u on the G_i -layer G_i^v is the vertex w of $V(G_i^v)$ which is the closest to u .

Suppose $D = \{G_1, \dots, G_k\}$ is a decomposition of an ordinary graph G . We say that two G_i -layers X_1 and X_2 are *adjacent* by G_j if and only if there exists an edge ab of a G_j -layer such that $a \in X_1$ and $b \in X_2$. In other words, the subgraph induced by the vertices of X_1 and X_2 is isomorphic to $G_i \square K_2$ where K_2 corresponds to the edge ab .

Let A and B be two ordinary graphs. The *greatest common divisor* of A and B is the graph X such that, for every three graphs W , Y , and Z with $A = W \square Y$ and $B = W \square Z$, X is a factor of W .

The goal of the rest of this section is to present useful results on the Cartesian product of undirected graphs.

One of the first results on the chromatic number of Cartesian products of undirected graphs is due to Sabidussi.

Theorem 5.4 (Sabidussi [168]). *For every two graphs G and H , we have:*

$$\chi(G \square H) = \max(\chi(G), \chi(H)).$$

Following this paper, Sabidussi proved one of the most important results on the Cartesian product: the unicity of the prime decomposition of connected graphs. This result was independently proved by Vizing.

Theorem 5.5 (Sabidussi [169] and Vizing [183]). *Every connected ordinary graph G admits a unique prime decomposition up to the order and isomorphisms of the factors.*

Using some arguments of [74] and the previous theorem, Imrich and Klavžar proved the following theorem.

Theorem 5.6 (Imrich and Klavžar [113, 114]). *If A , B and C are three ordinary graphs such that $A \square B = A \square C$, then $B = C$.*

These two theorems imply that the set of all graphs together with the disjoint union operation and the Cartesian product form a semi-ring with the cancellation property.

The unicity of the prime decomposition raises the question of the complexity of finding such a decomposition. The complexity of decomposition algorithms has been extensively

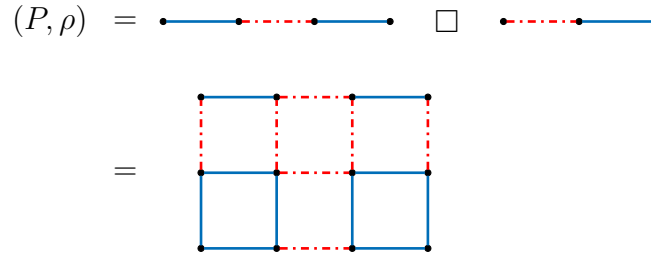


Figure 5.1: A signed graph (P, ρ) obtained as the Cartesian product of two signed paths.

studied. The first algorithm, by Feigenbaum, Hershberger and Schäffer [72] had a complexity of $O(n^{4.5})$ where n is the order of the graph (its size is denoted by m). In [195], Winkler proposed a different algorithm improving the complexity to $O(n^4)$. Then Feder [70] gave an algorithm in $O(mn)$ time and $O(m)$ space. The same year, Aurenhammer, Hagauer and Imrich [10] gave an algorithm in $O(m \log n)$ time and $O(m)$ space. The latest result is an optimal algorithm.

Theorem 5.7 (Imrich and Peterin [115]). *The prime factorization of connected ordinary graphs can be found in $O(m)$ time and space. Additionally a coordinate system can be computed in $O(m)$ time and space.*

5.2 Cartesian products of signed graphs

5.2.1 Definition

We recall the definition of the Cartesian product of signed graphs due to Germina, Hameed K. and Zaslavsky:

Definition 5.8 ([85]). Let (G, σ) and (H, π) be two signed graphs. The *Cartesian product* of (G, σ) and (H, π) , denoted by $(G, \sigma) \square (H, \pi)$, is the signed graph defined as follows:

- $V((G, \sigma) \square (H, \pi)) = V(G) \times V(H)$,
- the positive (resp. negative) edges are the pairs $\{(u_1, v_1), (u_2, v_2)\}$ such that:
 - $u_1 = u_2$ and $v_1 v_2$ is a positive (resp. negative) edge of (H, π) , or
 - $v_1 = v_2$ and $u_1 u_2$ is a positive (resp. negative) edge of (G, σ) .

See Figure 5.1 and Figure 5.2 for example of Cartesian products of signed graphs.

Note that the underlying graph of $(G, \sigma) \square (H, \pi)$ is the ordinary graph $G \square H$. From this definition, we can derive that the Cartesian product is associative and commutative. In particular $(\mathbb{S}\mathbb{G}, \uplus, \square, K_0, K_1)$ is a commutative semi-ring where $\mathbb{S}\mathbb{G}$ is the set of all signed graphs and \uplus is the disjoint union for signed graphs.

The following result shows that Cartesian products are compatible with homomorphisms of signed graphs and in particular with the switching operation.

Theorem 5.9. *If $(G, \sigma), (G', \sigma'), (H, \pi), (H', \pi')$ are four signed graphs such that $(G, \sigma) \xrightarrow{s} (G', \sigma')$ and $(H, \pi) \xrightarrow{s} (H', \pi')$, then:*

$$(G, \sigma) \square (H, \pi) \xrightarrow{s} (G', \sigma') \square (H', \pi').$$

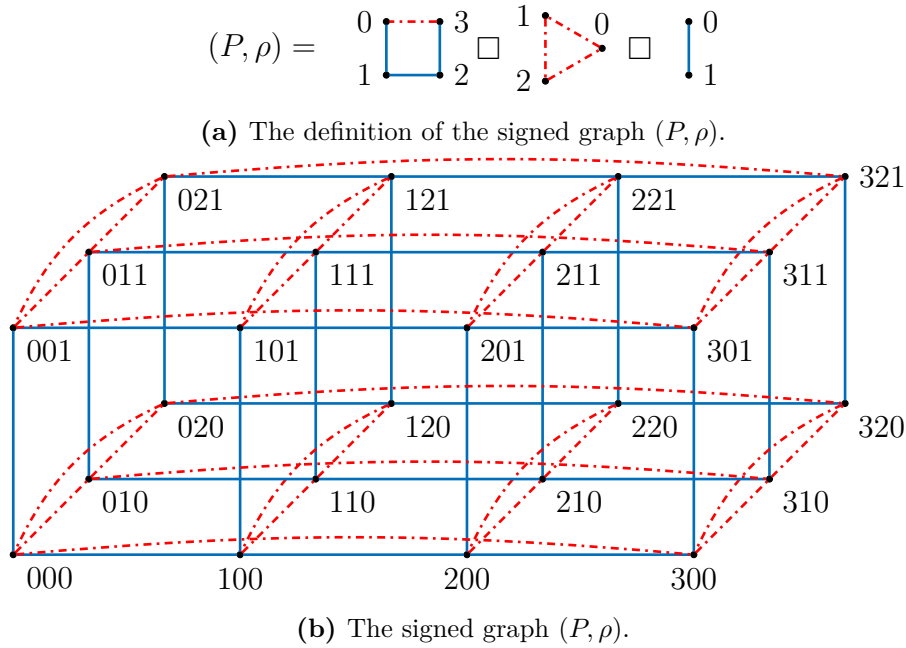


Figure 5.2: A signed graph (P, ρ) obtained as the Cartesian product of three signed graphs.

Proof. By commutativity of the Cartesian product and composition of homomorphisms, it suffices to show that $(G, \sigma) \square (H, \pi) \rightarrow_s (G', \sigma') \square (H, \pi)$. Since $(G, \sigma) \rightarrow_s (G', \sigma')$, there exists a set S of vertices and a homomorphism φ from G to G' such that if (G, σ_S) is the signed graph obtained from (G, σ) by switching the vertices of S , then $\sigma'(\varphi(e)) = \sigma_S(e)$ for every edge e of G . We note $P = (G, \sigma) \square (H, \pi)$ and $X = \{(g, h) \in V(G \square H) : g \in S\}$. Let P' be the signed graph obtained from P by switching the vertices in X .

If $(g, h)(g, h')$ is an edge of P , then in P' this edge was either switched twice if $g \in S$ or not switched if $g \notin S$. In both cases its sign did not change. If $(g, h)(g', h)$ is an edge of P , then in P' this edge was switched twice if $g, g' \in S$, switched once if $g \in S, g' \notin S$ or $g \notin S, g' \in S$, and not switched if $g, g' \notin S$. In each case its new sign is $\sigma_S(gg')$. Thus $P' = (G, \sigma_S) \square (H, \pi)$. Now define $\varphi_P(g, h) = (\varphi(g), h)$. It is a homomorphism from $G \square H$ to $G' \square H$ by definition. By construction, the target graph of φ_P is $(G', \sigma') \square (H, \pi)$ as the edges of H do not change and the target graph of φ is (G', σ') . \square

As mentioned before, we can derive the following corollary from Theorem 5.9.

Corollary 5.10. *If $(G, \sigma), (G, \sigma'), (H, \pi), (H, \pi')$ are four signed graphs such that $\sigma \equiv \sigma'$ and $\pi \equiv \pi'$, then:*

$$(G, \sigma) \square (H, \pi) \equiv (G, \sigma') \square (H, \pi').$$

From Theorem 5.9, and the fact that $(F, \sigma) \rightarrow_s K_2^+$ for every signed forest (F, σ) , we also get the following corollary:

Corollary 5.11. *If (G, σ) is a signed graph and (F, π) is a signed forest with at least one edge, then:*

$$\chi_s((G, \sigma) \square (F, \pi)) = \chi_s((G, \sigma) \square K_2^+).$$

In particular, for $n, m \geq 2$, $\chi_s((P_n, \sigma_1) \square (P_m, \sigma_2)) = 2$.

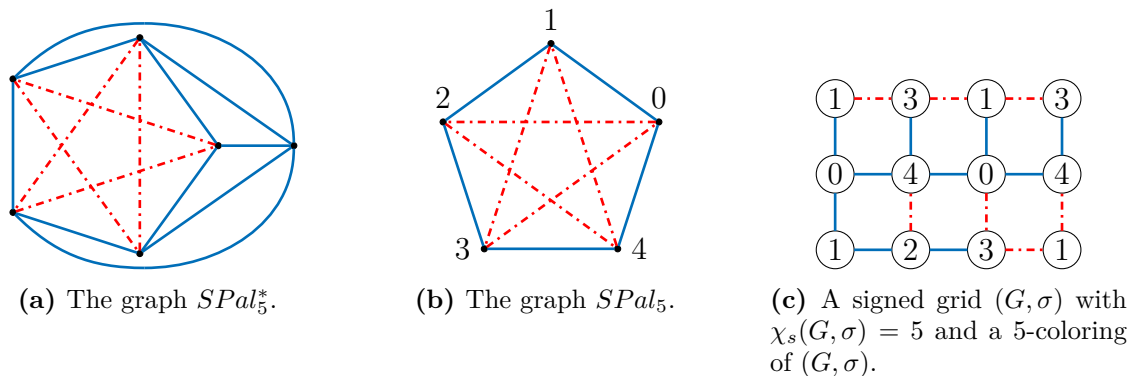


Figure 5.3: The signed graphs used in the proof of Theorem 5.12.

5.2.2 Signed grids

Note that there is a difference between considering the chromatic number of the Cartesian product of two signed graphs and the chromatic number of a signed graph whose underlying graph is a Cartesian product. For example, $C_4 = K_2 \square K_2$ but $4 = \chi_s(UC_4) \neq \chi_s(BC_4) = 2$. Another example comes from grid graphs: $\chi_s((P_n, \sigma_1) \square (P_m, \sigma_2)) = 2$, for any $n, m \in \mathbb{N}$, but the following theorem shows that not all signed grids have chromatic number 2.

Theorem 5.12. *If n and m are two integers with $1 \leq n \leq m$ and (G, σ) is a signed grid with $G = P_n \square P_m$, then $\chi_s(G, \sigma) \leq 6$. If $n \leq 4$, then $\chi_s(G, \sigma) \leq 5$. Moreover there exist signed grids with chromatic number 5.*

On our figures, we use dashed red edges to represent negative edges and solid blue edges for positive edges.

Proof. We will prove a more precise statement: every signed grid (G, σ) verifies $(G, \sigma) \rightarrow_s SPal_5^*$ where $SPal_5^*$ is the graph of Figure 5.3(a). This graph has the following (easy to check) property:

- (P) for every three vertices x, y, z of $SPal_5^*$, and every sign $\epsilon \in \{+1, -1\}$, if $x \neq z$ or $\epsilon = +1$, then there exists u and v in $SPal_5^*$, $u \neq v$, such that the closed walks $xyzu$ and $xyzv$ have sign ϵ .

To map (G, σ) to $SPal_5^*$, we will construct the homomorphism φ by induction. The vertex of G in line $i \in \{1, \dots, n\}$ and column $j \in \{1, \dots, m\}$ will be called $x_{i,j}$. Let $H_{i,j}$ be the subgraph of G induced by the vertices $x_{k,\ell}$ where $1 \leq k < i$, or $k = i$ and $1 \leq \ell \leq j$. We prove that for all i, j , $1 \leq i \leq n$ and $1 \leq j \leq m$, $H_{i,j} \rightarrow_s SPal_5^*$. It is easy to see that $H_{1,m} \rightarrow_s SPal_5^*$.

If $H_{i,m} \rightarrow_s SPal_5^*$ and $1 \leq i < n$, then $x_{i+1,1}$ has only one neighbor in $H_{i+1,1}$ and we can extend the previous homomorphism to $H_{i+1,1}$.

Suppose that φ is a homomorphism from $H_{i,j}$ to $SPal_5^*$, with $1 < i \leq n$ and $1 \leq j < m$. Let $C = x_{i,j+1}x_{i,j}x_{i-1,j}x_{i-1,j+1}$. If $C = BC_4$ or if $C = UC_4$ and $\varphi(x_{i,j}) \neq \varphi(x_{i-1,j+1})$, then we have two choices for $x_{i,j+1}$ by P (we might need to switch $x_{i,j}$) and we can extend the homomorphism to $H_{i,j+1}$. If $C = UC_4$ and $\varphi(x_{i,j}) = \varphi(x_{i-1,j+1})$, then these two vertices must be different. There were two possibilities for the choice of $\varphi(x_{i,j-1})$ in the previous step by P thus if we take the other one, we are back to the previous case where $\varphi(x_{i,j-1}) \neq \varphi(x_{i-1,j})$. Thus we can extend φ to $H_{i,j+1}$.

Hence, $H_{n,m} = (G, \sigma) \longrightarrow_s SPal_5^*$ which gives $\chi_s(G, \sigma) \leq 6$.

Suppose now that $n \leq 4$. If $n < 4$, then we could add one extra line to our grid to make it a $4 \times m$ grid. Therefore, we prove only the case where $n = 4$.

We construct a homomorphism $\varphi : (G, \sigma) \longrightarrow_s SPal_5$, column by column, where $SPal_5$ is the graph of Figure 5.3(b). The first column is a path and thus, we can map it arbitrarily to $SPal_5$. For a column with vertices $x_{1,j}, x_{2,j}, x_{3,j}, x_{4,j}$ and $1 < j \leq m$, we extend φ to the vertices of the column depending on the images of the vertices $x_{1,j-1}, x_{2,j-1}, x_{3,j-1}$ and $x_{4,j-1}$. Let v^{j-1} be the vector $(\varphi(x_{1,j-1}), \varphi(x_{2,j-1}), \varphi(x_{3,j-1}), \varphi(x_{4,j-1}))$. Note that any permutation of vertices of $SPal_5$ gives a homomorphism from $SPal_5$ to $SPal_5$ (which may require switching some vertices). Hence by symmetry between the vertices of $SPal_5$ and since φ does not map adjacent vertices to the same image, we can suppose that v^{j-1} is one of the following vectors: $(1, 2, 1, 2)$, $(1, 2, 1, 3)$, $(1, 2, 3, 1)$, $(1, 2, 3, 2)$, $(1, 2, 3, 4)$. Note that by inverting the order of the columns and permuting the colors, $(1, 2, 3, 2)$ and $(1, 2, 1, 3)$ are symmetrical. Hence we only have four cases to check for the images of the previous column.

Note that we need check that every BC_4 of (G, σ) is mapped to balanced closed walk in $SPal_5$ and that every UC_4 of (G, σ) is mapped to a UC_4 of $SPal_5$ for φ to be a homomorphism. Hence for each of the three following cycles: $C_1 = x_{1,j}x_{2,j}x_{2,j-1}x_{1,j-1}$, $C_2 = x_{2,j}x_{3,j}x_{3,j-1}x_{2,j-1}$ and $C_3 = x_{3,j}x_{4,j}x_{4,j-1}x_{3,j-1}$, we have two cases depending on whether the cycle is balanced or not. This represents 4×8 cases to check. We can reduce the number of cases as follows. If $\sigma(C_1) = +1$ then we can find an extension of φ by considering that $\sigma(C_1) = -1$ (*i.e.* C_1 is unbalanced) and replacing the image of $x_{1,j}$ in this extension by $\varphi(x_{2,j-1})$. The same can be done for the case where $\sigma(C_3) = +1$. Hence we only have two cases to consider depending on the sign of C_2 .

Let us give a possible way to extend φ in each remaining cases. For each case, we present an extension of φ to this column by giving the value of the vector $v^j = (\varphi(x_{1,j}), \varphi(x_{2,j}), \varphi(x_{3,j}), \varphi(x_{4,j}))$ corresponding to the extension.

Suppose first that $\sigma(C_2) = +1$. If $v^{j-1} = (1, 2, 1, 2)$, then we choose $v^j = (0, 4, 3, 0)$. If $v^{j-1} = (1, 2, 1, 3)$, then we choose $v^j = (3, 0, 2, 0)$. If $v^{j-1} = (1, 2, 3, 1)$, then we choose $v^j = (3, 0, 2, 4)$. If $v^{j-1} = (1, 2, 3, 4)$, then we choose $v^j = (3, 0, 2, 1)$. Now assume that $\sigma(C_2) = -1$. If $v^{j-1} = (1, 2, 1, 2)$, then we choose $v^j = (3, 0, 3, 0)$. If $v^{j-1} = (1, 2, 1, 3)$, then we choose $v^j = (4, 3, 4, 2)$. If $v^{j-1} = (1, 2, 3, 1)$, then we choose $v^j = (3, 0, 4, 0)$. If $v^{j-1} = (1, 2, 3, 4)$, then we choose $v^j = (0, 4, 1, 0)$. In each case, we can find a way to extend φ to the current column. Hence $(G, \sigma) \longrightarrow_s SPal_5$.

It is tedious but not difficult to check that the signed grid of Figure 5.3(c) cannot be mapped to a signed graph of order 4, thus its chromatic number is at least 5. In fact it is exactly 5. This concludes the proof. \square

Note that, independantly from our work, Dybizbański, Nenca and Szepietowski in [61] presented a computer assisted proof of the following result.

Theorem 5.13 (Dybizbański, Nenca and Szepietowski [61]). *If (G, σ) is a $n \times m$ signed grid with $n \leq 7$, then $\chi_s(G, \sigma) \leq 5$.*

They also prove the upper bound of 6 on every signed grid. Their result is thus better than Theorem 5.12. Nevertheless, we still do not know whether the upper bound for signed grids is 5 or 6.

Question 5.14. *What is the maximal value of $\chi_s(G, \sigma)$ when (G, σ) is a signed grid? Is it 5 or 6?*

5.2.3 Prime factor decomposition: existence and unicity

Our goal now is to prove that each connected signed graph has a unique prime s -decomposition. Let us start with some definitions.

Definition 5.15. A signed graph (G, σ) is said to be s -prime if and only if there do not exist two signed graphs (A, π_A) and (B, π_B) such that $(G, \sigma) \equiv (A, \pi_A) \square (B, \pi_B)$. An s -decomposition of a signed connected graph (G, σ) is a multiset of signed graphs $D = \{(G_1, \pi_1), \dots, (G_k, \pi_k)\}$ such that:

1. the (G_i, π_i) 's are signed graphs containing at least one edge and
2. $(G, \pi) \equiv (G_1, \pi_1) \square \dots \square (G_k, \pi_k)$.

An s -decomposition D is *prime* if all the (G_i, π_i) 's are s -prime. The (G_i, π_i) 's are called *factors* of D .

Let (G, σ) be a signed graph such that there exist two ordinary graphs A and B for which $G = A \square B$. A *signed A -layer* of (G, σ) is a signed subgraph (A^v, σ_{A^v}) of (G, σ) where A^v is an A -layer of G .

Let D be an s -decomposition of (G, σ) and D' be the decomposition of G corresponding to D by forgetting the signs. For a factor (A, π_A) of D , an (A, π_A) -*layer* of (G, σ) is a signed subgraph (A^v, π_{A^v}) of (G, σ) such that A^v is an A -layer of G for D' where A is the factor of D' corresponding to the factor (A, π_A) of D . By definition of (A, π_A) , $(A^v, \pi_{A^v}) \equiv (A, \pi_A)$.

Note that if $G = A \square B$, then it is not always true that (G, σ) is the Cartesian product of two signed graphs. For example, UC_4 is s -prime but C_4 is not a prime graph as $C_4 = K_2 \square K_2$. The following lemma tells us in which cases $(G, \sigma) \equiv (A, \pi_A) \square (B, \pi_B)$, and will be a useful tool for decomposing signed graphs.

Lemma 5.16. *If (G, σ) , (A, π_A) and (B, π_B) are three connected signed graphs with $G = A \square B$, then $(G, \sigma) \equiv (A, \pi_A) \square (B, \pi_B)$ if and only if:*

1. all signed A -layers are equivalent to (A, π_A) ,
2. at least one signed B -layer is equivalent to (B, π_B) , and
3. for each edge e of A and each pair of distinct copies e_1, e_2 of e , if e_1 and e_2 belong to the same signed 4-cycle, then this cycle is a BC_4 .

Note that, in the previous lemma, all signed B -layers are equivalent to (B, π_B) but we only need to verify that for one of them to conclude.

Proof. (\Rightarrow) This follows from the definition of the Cartesian product.

(\Leftarrow) We will do the following independent switchings: switch all signed A -layers to have the same signature π_A .

Now we claim that all signed B -layers have the same signature π'_B equivalent to π_B . Indeed take one edge xy of B and two copies of this edge x_1y_1 and x_2y_2 in G . Take a shortest path P from x_1 to x_2 in the A^{x_1} -layer. Now if u_1, u_2 are two consecutive vertices

along P and v_1 and v_2 are their projections on A^{y_1} , then $u_1u_2v_2v_1$ is a BC_4 in (G, σ) by the third hypothesis as u_1v_1 and u_2v_2 are copies of the edge xy .

As u_1u_2 and v_1v_2 have the same sign by the previous switchings, it must be that u_1v_1 and u_2v_2 have the same sign. Thus all copies of an edge of B have the same sign.

Hence, $(G, \sigma) \equiv (A, \pi_A) \square (B, \pi'_B) \equiv (A, \pi_A) \square (B, \pi_B)$ by Theorem 5.9. \square

One of our main results is the following Prime Decomposition Theorem.

Theorem 5.17 (Prime s -decomposition Theorem). *If (G, σ) is a connected signed graph and D is the prime decomposition of G , then (G, σ) admits a unique (up to isomorphism and order of the factors) prime s -decomposition D_s . Moreover, if we see D_s as a decomposition of G , then D is finer than D_s .*

For proving this theorem, we need the following lemma.

Lemma 5.18. *If (G, σ) is a connected signed graph that admits two prime s -decompositions D_1 and D_2 , then there are two signed graphs (X, π_X) and (Y, π_Y) such that $(G, \sigma) \equiv (X, \pi_X) \square (Y, \pi_Y)$ with $D_1 = \{(X, \pi_X)\} \cup D'_1$ and $D_2 = \{(X, \pi_X)\} \cup D'_2$, where D'_1 and D'_2 are two s -decompositions of (Y, π_Y) .*

Proof. Suppose there exists a signed graph (G, σ) that admits two s -decompositions D_1 and D_2 . Fix an edge e of (G, σ) which belongs to some Z -layer Z^e of the prime decomposition of G . The edge e belongs to some (A, π_A) -layer in D_1 and to some (B, π_B) -layer in D_2 . The graph Z is a factor of A and B by unicity of the prime factor decomposition of G . Let X be the greatest common divisor of A and B . Since $e \in E(Z^e)$, e is in some X -layer X^e . Now $G = X \square Y$ for some graph Y . Let us show that $(G, \sigma) \equiv (X, \pi_X) \square (Y, \pi_Y)$ for some signatures π_X and π_Y of X and Y , respectively. We can suppose that $Y \neq K_1$ and $A \neq B$, as otherwise the result is immediate.

First we want to show that all signed X -layers have equivalent signatures. Take two adjacent signed X -layers. If they are in different signed A -layers, then they are equivalent since they represent the same part of (A, π_A) . If they are in the same signed A -layer, then they are in different signed B -layers since X is the greatest common divisor of A and B . The same argument works in this case. Thus two adjacent signed X -layers are isomorphic to the same signed graph (X, π_X) , and since there is only one connected component in Y , all signed X -layers have equivalent signatures.

Let π_Y be the signature of one signed Y -layer. Fix e' an edge of X , and (X_1, π_{X_1}) , (X_2, π_{X_2}) two signed X -layers. Now consider the signed 4-cycle (if it exists) containing the copies of this edge in each of the two signed X -layers. If (X_1, π_{X_1}) and (X_2, π_{X_2}) are in different signed A -layers, then this signed cycle is a BC_4 by Lemma 5.16, otherwise this signed cycle is a BC_4 as (X_1, π_{X_1}) and (X_2, π_{X_2}) are in different signed B -layers by the same argument.

By Lemma 5.16, we can conclude that $(G, \sigma) \equiv (X, \pi_X) \square (Y, \pi_Y)$.

Now suppose that $A = X \square W$. Using Lemma 5.16, we can show that $(A, \pi_A) \equiv (X, \pi_X) \square (W, \pi_W)$. Indeed all signed X -layers have equivalent signatures since $(G, \sigma) \equiv (X, \pi_X) \square (Y, \pi_Y)$ and all signed 4-cycles between two copies of an edge of X are BC_4 by the same argument. As (A, π_A) is s -prime, this implies $(X, \pi_X) \equiv (A, \pi_A)$. Thus $(X, \pi_X) \equiv (A, \pi_A) \equiv (B, \pi_B)$ and this proves the lemma. \square

Proof of Theorem 5.17. Any signed graph (G, σ) has a prime s -decomposition by taking an s -decomposition that cannot be refined. Every prime s -decomposition of (G, σ) can

be considered as a decomposition of G , and the prime decomposition of G is finer than every such decomposition. We still have to show that the prime s -decomposition of (G, σ) is unique. Suppose, to the contrary, that (G, σ) is a minimal counter-example to the unicity. Thus (G, σ) has two prime s -decompositions D_1 and D_2 and, by Lemma 5.18, $(G, \sigma) \equiv (X, \pi_X) \square (Y, \pi_Y)$ with $D_1 = \{(X, \pi_X)\} \cup D'_1$ and $D_2 = \{(X, \pi_X)\} \cup D'_2$, where D'_1 and D'_2 are two s -decompositions of (Y, π_Y) . By minimality of (G, σ) , (Y, π_Y) has a unique prime s -decomposition, hence $D'_1 = D'_2$. Thus $D_1 = D_2$, a contradiction. \square

Note that Theorem 5.17 implies the following result.

Theorem 5.19. *If (A, π_A) , (B, π_B) and (C, π_C) are three signed graphs verifying $(A, \pi_A) \square (B, \pi_B) \equiv (A, \pi_A) \square (C, \pi_C)$, then $(B, \pi_B) \equiv (C, \pi_C)$.*

The proof of this result is exactly the same as the proof for ordinary graphs presented in [114]. Indeed, we have all the necessary tools used in the proof. The first one is Theorem 5.17, the other one is the semi-ring structure of signed graphs (quotiented by the equivalence relation) with the disjoint union and the Cartesian product which follows from the definition. See [114] for more details on the proof.

Another application of the prime s -decomposition theorem is to compute the frustration index (*i.e.* the minimum number of edges to remove from a signed graph to make it balanced) of a signed graph. Indeed, it is easy to see that the frustration index of a signed graph is equal to the product of the frustration indices of each of its factors. Hence given the prime s -decomposition of a signed graph, one can compute the frustration index of each prime factor and compute the frustration index of the whole graph.

5.2.4 Recognising Cartesian products of signed graphs

In the last part of this section, we propose an algorithm to decompose connected signed graphs. Decomposing a graph can be interpreted in multiple ways: finding a decomposition, identifying which edge of G belongs to which factor, or even better getting a coordinate system that is compatible with the decomposition. In [115], Imrich and Peterin gave an $O(m)$ time and space (m is the number of edges of G) algorithm for these three questions for ordinary graphs. More recently, in [116], they gave another algorithm in $O(m)$ time and space to decompose directed graphs.

Our goal is to give a similar algorithm for signed graphs based on their algorithm for directed graphs.

Theorem 5.20. *Let (G, σ) be a connected signed graph of order n and size m . We can find in time $O(m)$ and space $O(m)$ the prime s -decomposition of (G, σ) and a coordinate system for this decomposition.*

We take a coordinate system for a graph G corresponding to its prime decomposition D which can be computed in $O(m)$ time [115]. Let v be the vertex of G with coordinates all equal to zero. We order the vertices using a BFS traversal of the graph starting at v . If xy is an edge, then it is a down-edge (*resp.* up-edge, *resp.* cross-edge) of x when $d(v, x) < d(v, y)$ (*resp.* $d(v, x) > d(v, y)$, *resp.* $d(v, x) = d(v, y)$) where d denotes the distance in G . We proceed as described in Algorithm 1. We color the edges of G using the prime decomposition D of G : we associate to each factor X of D a color, which is then assigned to every edge belonging to an X -layer of G . We maintain a temporary

```

Input : A signed graph  $(G, \sigma)$ 
Output: The prime factor  $s$ -decomposition of  $(G, \sigma)$ 
1 Compute the prime factor decomposition  $D$  of  $G$ ;
2 Set the temporary decomposition of  $(G, \sigma)$  to be  $J = D$ ;
3  $Done \leftarrow \emptyset$ ;
4  $S \leftarrow \emptyset$ ;
5  $Treated \leftarrow \emptyset$ ;
6 forall vertices  $x$  taken according to a BFS ordering do
7   Add  $x$  to  $S$ ;
8   forall edges  $xy \notin Treated$  do
9     Determine the temporary color  $i$  of  $xy$  and  $J_i$  the current factor to which
       it belongs in the current decomposition;
10    Let  $x'y'$  be the projection of  $xy$  onto  $J_i^v$ ;
11    if  $xy$  and  $x'y'$  do not have the same sign and  $y \notin S$  then
12      Switch the vertex  $y$ ;
13      Add  $y$  to  $S$ ;
14    else if  $xy$  and  $x'y'$  have the same sign and  $y \notin S$  then
15      Add  $y$  to  $S$ ;
16    else if  $xy$  and  $x'y'$  do not have the same sign and  $y \in S$  then
17      Merge the temporary colors of all up-edges of  $y$  (and the temporary
       color of  $xy$ ) and update the decomposition;
18    end
19    Add  $xy$  to  $Treated$ ;
20  end
21  Add  $x$  to  $Done$ ;
22 end

```

Algorithm 1: A decomposition algorithm for signed graphs.

decomposition J of G for which we merge some factors, by means of recoloring the edges, during the algorithm. Our goal, at the end of the algorithm, is that $J = P$ where P is the prime s -decomposition of (G, σ) . We note $p_i(e)$ the projection of an edge $e = xy \in J_i^x$ to the temporary J_i -layer J_i^v .

First note that in Algorithm 1, the set $Done$ is not used. Therefore, it can be omitted. Its only purpose is to ease the correctness analysis of the algorithm. Let us make a few more remarks. The set $Done$ (resp. $Treated$) is used to record which vertex (resp. edge) has been processed by the algorithm. At any point of the execution of the algorithm, the set S corresponds to the set of vertices for which we have decided whether they need to be switched or not. If $x \in Done$ at some point of the algorithm then all its incident edges belong to the set $Treated$. Moreover, by construction of the BFS ordering, if xy is a down-edge of x in J_i^x , then for every vertex z , the projection $x'y'$ of xy on J_i^z is a down-edge of x' .

Claim 5.21. *After the merging in line 17 of the algorithm, v , y and x belong to the same layer.*

Proof. We just need to prove that y and v belong to the same layer after merging. Note that a layer J_i^a corresponds to all the vertices b which differ from a only by the i th

coordinate (in the current decomposition). Note also that the coordinate vector of a neighbor of y and the coordinate vector of y differ by only one coordinate. For any non-zero coordinate of y , there is an up-edge yz of y to a neighbor z of y which differs only on this non-zero coordinate (as the ordering is a BFS ordering and by the Cartesian product structure), therefore all factors J_ℓ corresponding to non-zero coordinates of y are merged. Hence, in this new coordinate system, y has at most one non-zero coordinate and thus y and v are in the same layer. \square

Claim 5.22. *Let ab and $a'b'$ be two edges of the set *Treated* at any moment of the algorithm. If $a'b' \in J_i^{a'}$ for some i and $p_i(a'b') = p_i(ab)$ (i.e. they represent the same edge of J_i), then ab and $a'b'$ have the same sign.*

Proof. By contradiction, suppose that $a'b'$ is the first edge such that, when added to *Treated*, there exists some edge $ab \in \textit{Treated}$ such that $p_i(a'b') = p_i(ab)$ and $a'b'$ and ab do not have the same sign. Let $a''b''$ be the edge $p_i(a'b')$. Note that no edge in *Treated* can change sign once it is into the set as both its endpoints are in S . By definition of $a'b'$, ab and $a''b''$ have the same sign since they both project to $a''b''$. Hence, it must be that $a'b'$ and $a''b''$ do not have the same sign.

Note that, $a'b'$ cannot be treated in the third if statement at line 16, as otherwise it would belong to some layer J_i^v after merging by Claim 5.21 and thus $a'b'$ would project to itself. Since $a'b'$ went through one of the first two if statements (lines 11 and 14), $a'b'$ and $a''b''$ have the same sign, a contradiction. \square

Proof of Theorem 5.20.

Correctness: First, let us show that J is finer than P , the prime s -decomposition of (G, σ) , at each step of the algorithm. It is true at the beginning of the algorithm by Theorem 5.17 as $J = D$. Suppose that J is finer than P at the beginning of step 8. In the if statement, if we enter the first two cases then we do not change J . Hence it is still finer than P at the end of the loop.

Suppose xy and $x'y'$ are not of the same sign and $y \in S$ (i.e. we enter line 17). As $y \in S$ and $xy \notin \textit{Treated}$, there is some neighbor z of y for which $z \in \textit{Done}$. We consider two cases depending on whether $z \in J_i^x$ or $z \notin J_i^x$.

Suppose first that $z \in J_i^x$.

Take a shortest path P_z in J_i^x from z to the projection p_v of v on J_i^x . All vertices of the path appear before z in the BFS ordering, thus all the edges of the path belong to the set *Treated*. The same holds for a shortest path P_x from p_v to x . In particular the walk W obtained by concatenating yz , P_z and P_x has all its edges in *Treated*. This implies that W and W' , its projection on J_i^v , have the same sign by Claim 5.22. Hence the closed walk C obtained by concatenating W with xy and its projection (W' concatenated with $x'y'$) have different signs and J_i^v and J_i^x do not have the same signature.

Let u be a neighbor of y such that uy is an up-edge of y and $u \notin J_i^x$. Every edge e' of the projection C' of C on J_i^u is in *Treated* as $d(v, e') < d(v, e)$ where e is the counterpart of e' in C (all vertices of C have an up-edge to their projection on J_i^u). In particular C and C' do not have the same sign and J_i^x and J_i^u do not have the same signature.

This implies that both layers are in the same factor of P . Indeed suppose that this is not the case. Then all signed cycles $abb'a'$, such that $ab \in J_i^x$ and $a'b'$ is its projection on J_i^u , must be BC_4 . For all edges ab of W , ab and $a'b'$ have the same sign by Claim 5.22, hence aa' and bb' also have the same sign (since the cycle is balanced). Now let $x''y''$

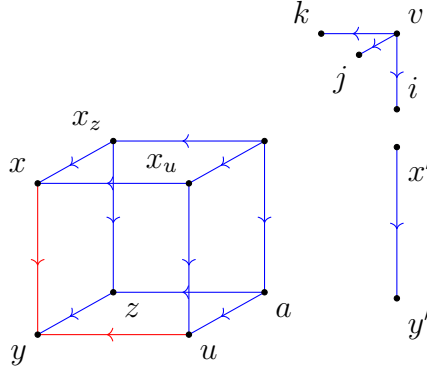


Figure 5.4: The second case of the correctness analysis. For simplicity, all edges which are in *Treated* are positive. The orientation of the edges represents the BFS order. The neighbors of v are labelled with the temporary color of their edge with v .

be the projection of xy on J_i^u . By going around W and by the previous observation, xx'' and yy'' have the same sign. Note that xy and $x''y''$ do not have the same sign as $x''y'' \in Treated$ ($x''y''$ has the same sign as $x'y'$). This implies that $xyy''x''$ is a UC_4 , a contradiction.

Hence we need to merge all temporary colors of all up-edges of y (including color i). Thus after this step J is still finer than P .

Suppose now that $z \notin J_i^x$ (see Figure 5.4).

In this case, z is the projection of y on J_i^z . Let x_z be the projection of x on J_i^z . Since yz is an up-edge of y , xx_z is an up-edge of x and $x_z \in Done$. Note that xx_z and yz have the same sign since both are in *Treated*. Also note that x_zz and $x'y'$ have the same sign since $x_zz \in Treated$. Hence $xyzx_w$ is a UC_4 . By the same arguments as before, these four vertices belong to the same signed factor of (G, σ) , hence we must merge i and, say j , the temporary colors of xy and yz respectively.

Let u be a neighbor of y such that uy is an up-edge of y of temporary color $k \notin \{i, j\}$. Let x_u be the projection of x on J_i^u . Note that x_uu and $x'y'$ have the same sign as $d(x_u, v) < d(x, v)$ (i.e. $x_u \in Done$). If xx_u and yu have the same sign, we have a UC_4 and must merge the temporary colors i and k . Suppose they have different signs. Note that y and z (resp. u) differ only by their j th coordinate (resp. k th coordinate). Let a be the vertex with the same coordinate as u except for its k th coordinate which is equal to the k th coordinate of z (see Figure 5.4). Note that a appears before z and u in the BFS ordering. Since the vertex a is a neighbor of z and u , both edges za and ua are down-edges of a . Hence $za \in Treated$ and za has the same sign as xx_u which is different from the sign of uy , and yz and ua also have the same sign since both are in *Treated*. In particular $yuaz$ is a UC_4 and these four vertices must be in the same factor of P . This implies that we must merge the temporary colors j and k which implies merging i and k .

At the end J is finer than P and J is an s -decomposition by Claim 5.22. Hence $J = P$.

Complexity: Due to the similarity of our algorithm with the one in [116], most of the complexity arguments given in [116] are still valid for our algorithm. The only differences between the two algorithms are the presence of the three sets *Done*, *S* and *Treated*, two more if blocks and the need to switch at some vertices. Let us address these three points. Each set can be encoded by a boolean in the data structure of vertices/edges. The second for loop checks each edge xy twice, once for each endpoint, but this still amounts to a

$O(m)$ iteration of the loop. The two additional if blocks are a $O(1)$ overhead for each iteration of the loop. The switch operation is another $O(m)$ total overhead as each edge can be switched at most once thanks to the presence of the set S . Hence the algorithm runs in time $O(m)$. The reader can find more details in [116], and in particular, how to compute the projections in constant time. \square

Note that this algorithm not only computes the prime s -decomposition of (G, σ) but finds a signature $\sigma' \equiv \sigma$ for which all layers of the Cartesian products have the same signature as their corresponding factors.

5.3 Chromatic number of Cartesian products of complete signed graphs and upper bounds

In this section, we show a simple upper bound on the chromatic number of a Cartesian product of two signed graphs and compute the chromatic number of some special complete signed graphs. We start by defining a useful tool on signed graphs.

5.3.1 s -redundant sets

In what follows we define the notion of an s -redundant set in a signed graph. Intuitively, if S is an s -redundant set of (G, σ) and x and y are two vertices that cannot be mapped to a same vertex by any homomorphism from (G, σ) , then they cannot be mapped to the same vertex by a homomorphism from $(G, \sigma) - S$.

Definition 5.23. Let (G, σ) be a signed graph and $S \subseteq V(G)$. We say that the set S is s -redundant if and only if, for every $x, y \in V(G) - S$ such that $xy \notin E(G)$, every $z \in S$ and every signature σ' with $\sigma' \equiv \sigma$, if $xzy = UP_3$ in (G, σ') , then there exists $w \in V(G) - S$ such that $xwy = UP_3$ in (G, σ') .

The following proposition provides an alternative formulation of the definition which is useful in order to prove that a set is an s -redundant set.

Proposition 5.24. *If (G, σ) is a signed graph and $S \subseteq V(G)$, then S is s -redundant if and only if for every $z \in S$, and every $x, y \in N(z) - S$ with $xy \notin E(G)$, there exists $w \in V(G) - S$ such that $xwyz$ is a BC_4 .*

Proof. Take $x, y \in V(G) - S$ such that $xy \notin E(G)$ and $z \in S$. If $xzy = UP_3$ in a signature $\sigma' \equiv \sigma$, then $x, y \in N(z)$. Now if S is an s -redundant set, then with the notation of the definition $xzyw$ is a BC_4 in (G, σ') and thus in (G, σ) . If $xzyw$ is a BC_4 and xzy is a UP_3 in a given signature σ' , then xwy is also a UP_3 as $xzyw$ is balanced. This proves the equivalence between the two statements. \square

The next theorem is the reason why we defined this notion. It allows us to compute an upper bound on the chromatic number of a signed graph as a function of the chromatic number of one of its subgraphs. One example of utilisation of this notion is given by the proof of Theorem 5.27.

Theorem 5.25. *If (G, σ) is a signed graph and S is an s -redundant set of (G, σ) , then*

$$\chi_s(G, \sigma) \leq |S| + \chi_s((G, \sigma) - S).$$

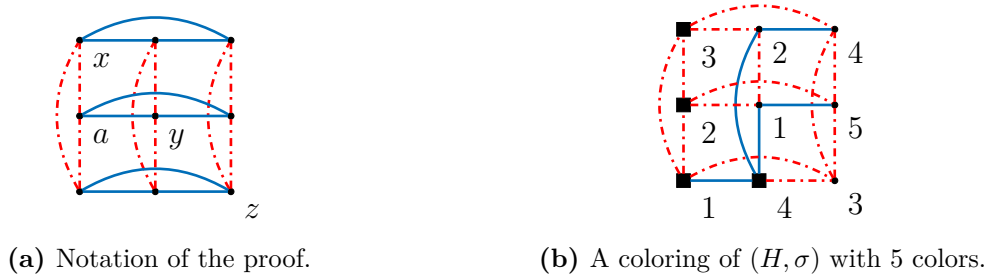


Figure 5.5: The signed graph $(H, \sigma) = K_3^+ \square K_3^-$ of Theorem 5.27. The big squared vertices have been switched.

Proof. Let c be a coloring of a signed graph $(G, \sigma') - S$ with $\chi_s((G, \sigma) - S)$ colors where $(G, \sigma') \equiv (G, \sigma)$. We define the coloring c' of (G, σ') as follows: $c'(v) = c(v)$ when $v \notin S$ and $c'(v)$ is a new color when $v \in S$. Hence c' uses at most $|S| + \chi_s((G, \sigma) - S)$ colors.

It is left to show that it is indeed a coloring of (G, σ') . As c is a coloring, c' does not assign the same color to two adjacent vertices. Suppose, by contradiction, that there exists two edges xy and $x'y'$ of opposite sign such that $c'(x) = c'(x')$ and $c'(y) = c'(y')$. As c is a coloring, all four vertices cannot be in $G - S$. W.l.o.g. suppose that $x \in S$. By definition of c' , $x' = x$, $y, y' \notin S$ and xyy' is a UP_3 in (G, σ') . As S is an s -redundant set, there exists $w \notin S$ such that ywy' is a UP_3 in $(G, \sigma') - S$. This contradicts the fact that c is a coloring of $(G, \sigma') - S$. \square

This result does not hold for any set S . For example, if $(G, \sigma) = UC_4$ and $S = \{v\}$ is a single vertex of G , then $\chi_s(G, \sigma) = 4$ and $\chi_s((G, \sigma) - v) = 2$.

5.3.2 Back to Cartesian products of complete signed graphs

As a direct corollary of Theorem 5.9, we get the following upper bound on the chromatic number of a Cartesian product of signed graphs.

Corollary 5.26. *If $(G_1, \sigma_1), \dots, (G_k, \sigma_k)$ are k signed graphs, then:*

$$\chi_s((G_1, \sigma_1) \square \dots \square (G_k, \sigma_k)) \leq \prod_{1 \leq i \leq k} \chi_s(G_i, \sigma_i).$$

We consider the Cartesian product of balanced and antibalanced complete graphs in our next result. Recall that K_p^+ (resp. K_q^-) is the complete graph with only positive edges (resp. negative edges).

Theorem 5.27. *For every two integers p, q with $p, q \geq 2$, we have*

$$\chi_s(K_p^+ \square K_q^-) = \left\lceil \frac{pq}{2} \right\rceil.$$

Proof. Let us note $(P, \pi) = K_p^+ \square K_q^-$. By symmetry between the sets of positive and negative edges, we can suppose $p \geq q$. First let us show that $\chi_s(P, \pi) \geq \left\lceil \frac{pq}{2} \right\rceil$.

If $q = 2$, then (P, π) is balanced and thus $\chi_s(P, \pi) = \chi(K_p \square K_2) = p$.

Suppose now that $p \geq q \geq 3$ and that $\chi_s(P, \pi) < \left\lceil \frac{pq}{2} \right\rceil$. Let φ be an optimal homomorphism from (P, π) . By the pigeon hole principle, there exist x, y and z three vertices of

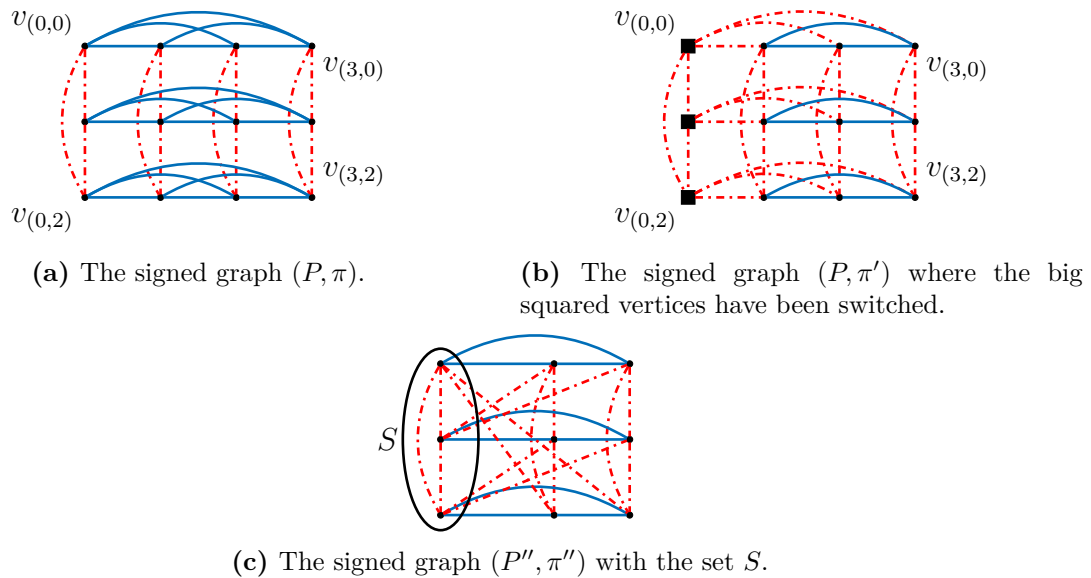


Figure 5.6: The signed graphs (P, π) , (P, π') and (P'', π'') of Theorem 5.27 when $(P, \pi) = K_4^+ \square K_3^-$.

the Cartesian product with the same image by φ . They belong to three distinct K_p^+ -layers and three distinct K_q^- -layer as these are complete graphs. Consider the subgraph (H, σ) of (P, π) composed of vertices which are in the same K_p^+ -layers as one of x, y, z and in the same K_q^- -layers as one of x, y and z . We have $(H, \sigma) = K_3^+ \square K_3^-$ (see Figure 5.5(a)).

By assumption x, y and z of (H, σ) are identified by φ (possibly after switching some of them). By the pigeon hole principle, two of x, y and z are both switched or both non-switched. Without loss of generality suppose they are x and y . Then if a is one of their common neighbors in H , the edges xa and ya are of different signs, thus x and y cannot be identified. This is a contradiction.

We now prove that $\chi_s(P, \pi) \leq \lceil \frac{pq}{2} \rceil$ by induction. If $p = 2$, then $(P, \pi) \equiv BC_4$ and $\chi_s(P, \pi) = 2 \leq 2$. If $p = 3$ and $q = 2$, then $(P, \pi) \equiv BC_3 \square K_2$ whose chromatic number is 3. If $p = 3$ and $q = 3$, then $(P, \pi) \equiv K_3^+ \square K_3^-$. In this case, we have $\chi_s(P, \pi) = 5$, as Figure 5.5(b) gives a 5-coloring of (P, π) .

Now we can assume $p \geq 4$. Let $V(P) = \{v_{(i,j)}, 0 \leq i < p, 0 \leq j < q\}$ such that for every i , the set $\{v_{(i,j)}\}_{0 \leq j < q}$ induces a negative complete graph and for every j , the set $\{v_{(i,j)}\}_{0 \leq i < p}$ induces a positive complete graph (see Figure 5.6(a)). Now switch all vertices in $\{v_{(i,j)} : i = 0\}$ to obtain the signed graph (P, π') (see Figure 5.6(b)) and then identify $v_{(0,j)}$ with $v_{(1,j+1)}$ (which are non-adjacent) for every $j \in \llbracket 0, q-1 \rrbracket$, where indices are taken modulo q , to obtain the graph (P'', π'') (see Figure 5.6(c)). Let S be the set of identified vertices in (P'', π'') . We want to show that S is s -redundant in order to use the induction hypothesis. Take $z \in S$ and $x, y \in N(z) - S$ such that $xy \notin E(P'')$. If xzy is an unbalanced path of length 2, then x is some $v_{(i,j)}$ and y is some $v_{(k,j+1)}$ with $i, k \geq 2$. For $a = v_{(i,j+1)}$, $xayz$ is a BC_4 .

By Proposition 5.24, S is s -redundant and thus

$$\chi_s(P, \pi) \leq \chi_s(P'', \pi'') \leq |S| + \chi_s((P'', \pi'') - S)$$

by Theorem 5.25. By induction hypothesis, as $(P'', \pi'') - S = K_{p-2}^+ \square K_q^-$, we get

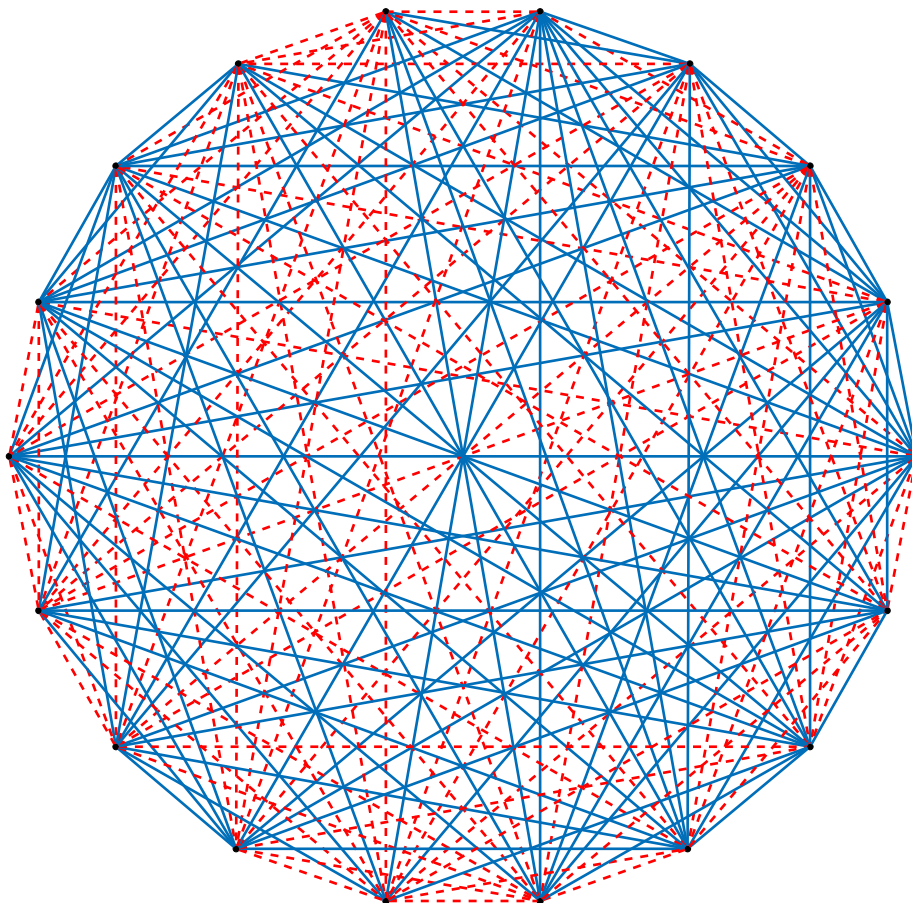


Figure 5.7: A signed graph K of order 18 such that $\chi_s(K \square K_2) = 25$.

$\chi_s((P'', \pi'') - S) \leq \left\lceil \frac{(p-2)q}{2} \right\rceil$. Thus $\chi_s(P, \pi) \leq q + \left\lceil \frac{pq}{2} \right\rceil - q \leq \left\lceil \frac{pq}{2} \right\rceil$. \square

For the Cartesian product $K_p^+ \square K_p^-$, the upper bound of Corollary 5.26 is p^2 while we proved in Theorem 5.27 that the chromatic number is $\left\lceil \frac{p^2}{2} \right\rceil$. We thus have an example where the chromatic number is greater than half the simple upper bound.

Question 5.28. *What is the supremum of the set of real numbers $\lambda \in [\frac{1}{2}, 1]$ such that there exist signed graphs $(G_1, \sigma_1), \dots, (G_k, \sigma_k)$, each with at least one edge, such that:*

$$\chi_s((G_1, \sigma_1) \square \dots \square (G_k, \sigma_k)) \leq \lambda \prod_{1 \leq i \leq k} \chi_s(G_i, \sigma_i)?$$

In Figure 5.7, we have an example of a graph K such that $K \square K_2$ has chromatic number 25 (checked by computer). The ratio between the chromatic number and the upper bound is $\frac{25}{36} = 0.69444$. It is the largest ratio we have found by randomly sampling bigger and bigger complete signed graphs. This leads us to believe that the following conjecture holds.

Conjecture 5.29. *For every fixed $\varepsilon > 0$, there exist signed graphs $(G_1, \sigma_1), \dots, (G_k, \sigma_k)$, with each at least one edge, such that:*

$$\chi_s((G_1, \sigma_1) \square \dots \square (G_k, \sigma_k)) \geq (1 - \varepsilon) \cdot \prod_{1 \leq i \leq k} \chi_s(G_i, \sigma_i).$$

5.4 Chromatic number of Cartesian products of signed cycles

The goal of this section is to determine the chromatic number of the Cartesian product of two signed cycles. As there are four kind of cycles (balanced/unbalanced and even/odd length), we have a number of cases to analyse. In most cases some simple observations are sufficient to conclude. For the other cases, we need the following lemma whose proof is given in subsections 5.4.1 to 5.4.6, due to its length.

Lemma 5.30. *For every two integers $p, q \in \mathbb{N}$ such that $p \geq 1$ and $q \geq 3$:*

$$\chi_s(BC_{2p+1} \square UC_q) > 4.$$

With this lemma, we can state the main result of this section.

Theorem 5.31. *If (C_1, σ) and (C_2, σ_2) are two signed cycles, then the chromatic number of $(P, \pi) = (C_1, \sigma_1) \square (C_2, \sigma_2)$ is given by Table 5.1, depending on the types of (C_1, σ_1) and (C_2, σ_2) .*

$(C_1, \sigma_1) \square (C_2, \sigma_2)$	BC_{even}	BC_{odd}	UC_{even}	UC_{odd}
BC_{even}	2	3	4	3
BC_{odd}	3	3	5	5
UC_{even}	4	5	4	5
UC_{odd}	3	5	5	3

Table 5.1: The chromatic number of Cartesian products of signed cycles.

Proof. If G is a cycle of type BC_{even} (resp. $BC_{odd}, UC_{even}, UC_{odd}$), then $G \rightarrow_s BC_2 = K_2$ (resp. BC_3, UC_4, UC_3). By computing the chromatic numbers of the Cartesian products of (G, σ) and (H, π) when they belong to $\{K_2, BC_3, UC_4, UC_3\}$, we get an upper bound for each of the Cartesian product type equal to the corresponding value in the table. These cases, up to symmetry between the sets of positives and negatives edges, are represented in Figure 5.8. Note that to color some graphs, we switched some vertices.

For the lower bound, note that $\chi_s((C_1, \sigma_1) \square (C_2, \sigma_2)) \geq \max(\chi_s(C_1, \sigma_1), \chi_s(C_2, \sigma_2))$. Theorem 2.20 concludes for the cases where the chromatic number is at most 4. Lemma 5.30 allows us to conclude for the remaining cases as $\chi_s(BC_{2p+1} \square UC_q) = \chi_s(UC_{2p+1} \square UC_q)$ by symmetry between the two edge signs. \square

One further question would be to compute the chromatic number of the Cartesian product of an arbitrary number of signed cycles. Note that if $H \in \{K_2, BC_3, UC_3, UC_4\}$, then $H \square H \rightarrow_s H$. This implies that, for these four graphs, it is only interesting to look at Cartesian products of the form $K_2^a \square BC_3^b \square UC_4^c \square UC_3^d$ where $a, b, c, d \in \{0, 1\}$. Moreover, we can suppose that $a = 0$ if one of b, c or d is non-zero. Thus the only interesting case left to solve is determining the value of $\chi_s(BC_3 \square UC_3 \square UC_4)$.

To extend this to any length, using the same argument as in Theorem 5.31, would require that we obtain a lower bound for $\chi_s(BC_{2p+1} \square UC_{2q+1} \square UC_{2r})$ which is equal to $\chi_s(BC_3 \square UC_3 \square UC_4)$.

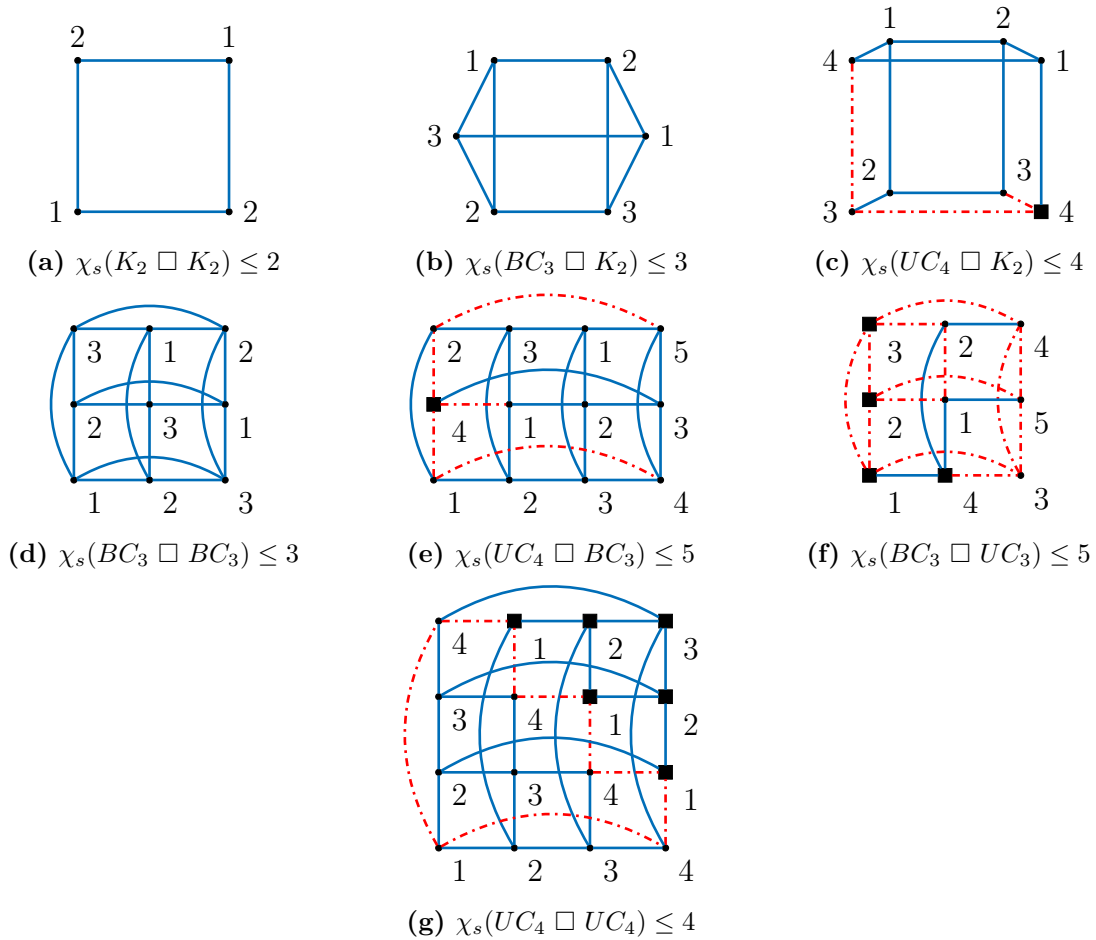


Figure 5.8: Coloring of Cartesian products of signed cycles. The large squared vertices have been switched in the Cartesian product.

5.4.1 Signed complete graphs of order 4

We count the number of signed complete graphs on four vertices. This result will be useful in the proof of Lemma 5.30.

Theorem 5.32. *There are three complete signed graphs of order 4 (see Figure 5.9). They are the signed graph $K_4^+ = (K_4, \emptyset)$ with only positive edges, the signed graph $K_4^- = (K_4, E(K_4))$ with only negative edges and the signed graph $K_4^{mixed} = (K_4, \{ab\})$ where a and b are two vertices of K_4 .*

Proof. Let (K_4, σ) be a complete signed graph on four vertices. Arbitrarily choose u to be one of the vertices of (K_4, σ) . By switching the neighbors of u if needed, we can suppose that u is only incident to positive edges. Let x, y, z be the other three vertices of (K_4, σ) . If the triangle xyz is all-positive, then $(K_4, \sigma) = K_4^+$, if the triangle is all-negative, then by switching u , we get $(K_4, \sigma) = K_4^-$. If the triangle has only one negative edge, then $(K_4, \sigma) = K_4^{mixed}$. Otherwise, the triangle has two negative edges, by switching the vertex with the two negative edges, we get $(K_4, \sigma) = K_4^{mixed}$. \square

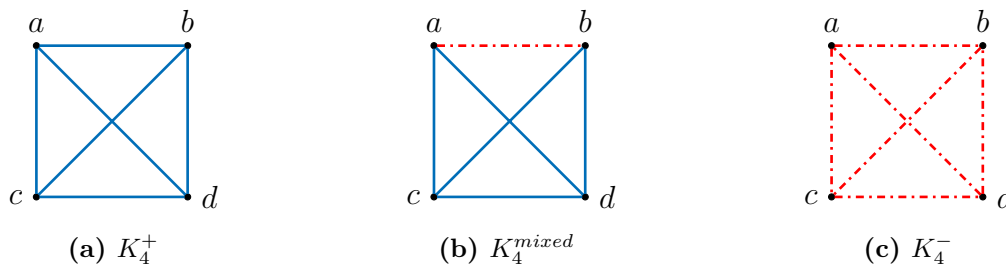


Figure 5.9: The three complete signed graphs of order 4

5.4.2 Beginning of the proof of Lemma 5.30

Our goal is to prove Lemma 5.30. For that, take some integers p and q such that $p \geq 1$ and $q \geq 3$, let $(P, \pi) = BC_{2p+1} \square UC_q$, and suppose that, by absurd, $\chi_s(P, \pi) \leq 4$.

Claim 5.33. *We have $(P, \pi) \rightarrow_s K_4^{mixed}$.*

Proof. Since $\chi_s(P, \pi) \leq 4$, $(P, \pi) \rightarrow_s (K_4, \rho)$ for some signature ρ of K_4 .

Every equivalent signature of BC_{2p+1} has at least one positive edge. Similarly, every equivalent signature of UC_q has at least one negative edge. Thus, in every equivalent signature of (P, π) , there is at least one positive edge and one negative edge. So (H, ρ) cannot be (K_4, \emptyset) nor $(K_4, E(K_4))$. By Theorem 5.32, since there are only three complete signed graphs of order 4, (H, ρ) is K_4^{mixed} . \square

From now on, we suppose that we fixed a homomorphism φ from (P, π) to K_4^{mixed} . We label the vertices of K_4^{mixed} as in Figure 5.9(b). Therefore, there exists a signed graph $(P, \pi') \equiv (P, \pi)$ for which $v \mapsto \varphi(v)$ is a coloring.

The proof of Lemma 5.30 is divided into four parts. First, by considering the graph P as a toroidal grid, we define what we mean for a walk to make a “turn” around the torus in subsection 5.4.3. Then, by considering the coloring of (P, π') corresponding to φ and the connected components of (P, π') induced by colors a and b , we link the number of “crossings” of some boundaries of the components with a vertical (or horizontal) cycle and the number of ab edges of this cycle in subsection 5.4.4. In subsection 5.4.5, we connect this number of ‘crossings’ to the number of turns and we conclude the proof in subsection 5.4.6.

5.4.3 Number of turns in P

The goal of this subsection is twofold. First, we want to establish another definition of P as a toroidal grid *i.e.* the quotient of some infinite grid. Secondly, we want to define the quantities $\tau_x(W)$ and $\tau_y(W)$ for each closed walk W of T . They represent the number of turns in each direction of the torus made by the closed walk W .

If G is a graph with vertex set a group H and Q is a subgroup of H , then the *quotient graph* G/Q over the vertices H/Q is defined by identifying the vertices in the same equivalence class. Similarly, if $W = s_0, \dots, s_n$ is a walk on G , then the *quotient walk* W' on G/Q is the sequence $\bar{s}_0, \dots, \bar{s}_n$.

Definition 5.34. We can associate with \mathbb{Z}^2 an infinite graph G^∞ whose vertex set $V(G^\infty)$ is the set $\{v_{x,y} : (x, y) \in \mathbb{Z}^2\}$ and whose edge set is the set of pairs $\{v_{x,y}v_{x',y'}\}$, where either

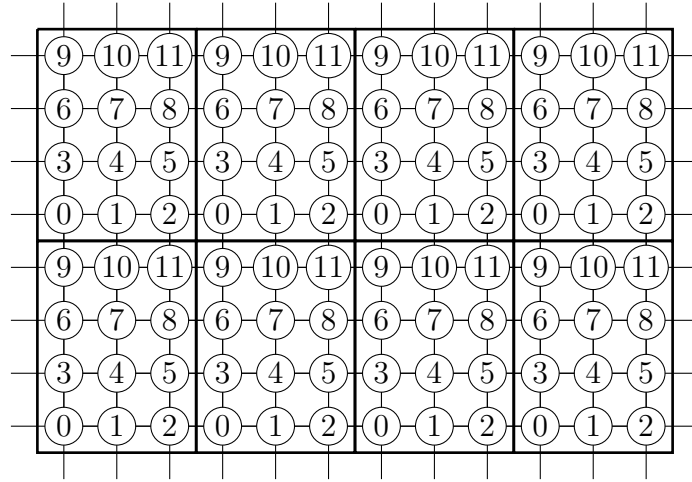


Figure 5.10: A subgraph of the graph G . Vertices with the same label are identified in P . Here $q = 4$ and $2p + 1 = 3$.

$x = x'$ and $|y - y'| = 1$, or $y = y'$ and $|x - x'| = 1$. We can then redefine the graph P as the quotient G^∞ / Q where $Q = \mathbb{Z}_{2p+1} \times \mathbb{Z}_q$. In other words take the graph G^∞ where we identify each vertex $v_{x,y}$ with $v_{x',y'}$ when $x - x'$ is a multiple of $2p + 1$ and $y - y'$ is a multiple of q . The graph G^∞ can be seen as an unfolding of the toroidal grid P . Figure 5.10 represents a subgraph of G^∞ when $q = 4$ and $2p + 1 = 3$. An edge of G^∞ of the form $v_{u,w}v_{u+i,w}$ (resp. $v_{u,w}v_{u,w+i}$) for $i \in \{-1, 1\}$, is an *horizontal edge* (resp. *vertical edge*) of G^∞ . An edge e of P is an *horizontal edge* (resp. *vertical edge*) if it is the quotient of horizontal (resp. vertical) edges of G^∞ .

Definition 5.35. Let W_{G^∞} be a walk in G^∞ and W_P a walk in P . We say that W_{G^∞} is a *representation* of W_P if and only if $W_{G^\infty} / Q = W_P$. We also say that W_{G^∞} *represents* W_P .

By definition, all representations of W_P have the same number of vertices as W_P . Let us make the following observation on the representations of a walk W_P .

Observation 5.36. If $W_{G^\infty}^1 = (s_i^1)_{0 \leq i \leq n}$ and $W_{G^\infty}^2 = (s_i^2)_{0 \leq i \leq n}$ are two walks (of the same length) in G^∞ representing W_P , then there exist $\alpha, \beta \in \mathbb{Z}$ such that for all $i \in \{0, \dots, n\}$, if $s_i^1 = v_{x,y}$, then $s_i^2 = v_{x+\alpha(2p+1), y+\beta q}$. In particular, if they have the same first vertices, then $W_{G^\infty}^1 = W_{G^\infty}^2$.

We are now ready to define what is a turn of a walk around the torus.

Definition 5.37. Let W_{G^∞} be a $(v_{x,y}, v_{z,t})$ -walk in G^∞ . We define the *number of horizontal turns* τ_x and the *number of vertical turns* τ_y of W_{G^∞} by:

$$\tau_x(W_{G^\infty}) = \left\lfloor \frac{z - x}{2p + 1} \right\rfloor, \tau_y(W_{G^\infty}) = \left\lfloor \frac{t - y}{q} \right\rfloor.$$

For a closed walk W_P in P , let $\tau_x(W_P) = \tau_x(W_{G^\infty})$ (resp. $\tau_y(W_P) = \tau_y(W_{G^\infty})$) be the number of horizontal (resp. vertical) turns of W_P where W_{G^∞} is an arbitrary representation of W_P .

Claim 5.38. For every closed walk W_P , the two quantities $\tau_x(W_P)$ and $\tau_y(W_P)$ are integers and do not depend on the choice of the representation W_{G^∞} of W_P .

Proof. First if W_P is a closed walk in P and W_{G^∞} represents W_P , then $\overline{v_{x,y}} = \overline{v_{z,t}}$ thus $z = x + n(2p + 1)$ and $t = y + mq$ for some integers $n, m \in \mathbb{Z}$. Hence $\tau_x(W_{G^\infty})$ and $\tau_y(W_{G^\infty})$ are integers.

Now take two representations $W_{G^\infty}^1$ and $W_{G^\infty}^2$ of W_P . By Observation 5.36, if $W_{G^\infty}^1$ is a $(v_{x_1, y_1}, v_{z_1, t_1})$ -walk and $W_{G^\infty}^2$ is a $(v_{x_2, y_2}, v_{z_2, t_2})$ -walk, then $x_2 = x_1 + \alpha(2p + 1)$, $y_2 = y_1 + \beta q$, $z_2 = z_1 + \alpha(2p + 1)$ and $t_2 = t_1 + \beta q$. Thus $\tau_x(W_{G^\infty}^1) = \tau_x(W_{G^\infty}^2)$ and $\tau_y(W_{G^\infty}^1) = \tau_y(W_{G^\infty}^2)$. Hence this quantity is well defined for W_P . \square

The main result of this subsection is the following proposition.

Proposition 5.39. *If W_P is a closed walk in P of even length, then:*

$$q\tau_y(W_P) + \tau_x(W_P) \equiv 0 \pmod{2}.$$

Proof. Let W_{G^∞} be a $(v_{x,y}, v_{z,t})$ -walk which is a representation of W_P in G^∞ . For each horizontal (resp. vertical) edge e of the form $v_{u,w}v_{u+i,w}$ (resp. $v_{u,w}v_{u,w+i}$) for $i \in \{-1, 1\}$, let $\ell(e) = i$. Let $E_h(W_{G^\infty})$ be the set of horizontal edges of W_{G^∞} and $E_v(W_{G^\infty})$ the set of vertical edges of W_{G^∞} . We then have:

$$\sum_{e \in E_h(W_{G^\infty})} \ell(e) \equiv z - x \equiv (2p + 1)\tau_x(W_P) \equiv \tau_x(W_P) \pmod{2},$$

and

$$\sum_{e \in E_h(W_{G^\infty})} \ell(e) \equiv \sum_{e \in E_h(W_{G^\infty})} 1 \equiv |E_h(W_{G^\infty})| \pmod{2}.$$

Similarly,

$$|E_v(W_{G^\infty})| \equiv t - y \equiv q\tau_y(W_P) \pmod{2}.$$

As W_P and W_{G^∞} are of even length, we get:

$$0 \equiv |E(W_{G^\infty})| \equiv q\tau_y(W_P) + \tau_x(W_P) \pmod{2}.$$

\square

5.4.4 Regions induced by a coloring of (P, π)

The aim of this section is to define a suitable set of walks in order to apply Proposition 5.39. For this, we will introduce several topological notions.

Definition 5.40. Let $P_{AB} = P[\varphi^{-1}\{a, b\}]$ and $P_{CD} = P[\varphi^{-1}\{c, d\}]$, the subgraphs of P induced by the vertices colored a and b and by the vertices colored c and d , respectively. A *region* X of P is a connected component of P_{AB} or P_{CD} . We say that X is *of type ab* in the first case and *of type cd* in the latter. The *boundary* ∂X of a region X is the subset of vertices of X that are adjacent to a vertex not in X :

$$\partial X = \{x \in X : N(x) \not\subseteq X\}.$$

The configuration of Figure 5.11 is called the *flat border configuration*. That is to say, for a region X , this configuration consists in two vertices $x, y \in X$ and two vertices $w, z \notin X$ such that xy, xz, yw and wz belong to $E(P)$.

Claim 5.41. *The flat border configuration does not appear in the coloring of (P, π') .*

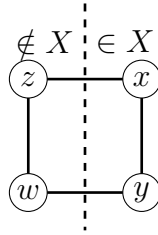


Figure 5.11: The flat border configuration when $x, y \in X$ and $z, w \notin X$, where X is a region.

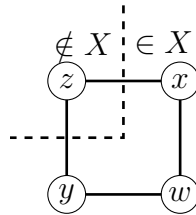


Figure 5.12: The vertices x and y of the region X are border neighbors.

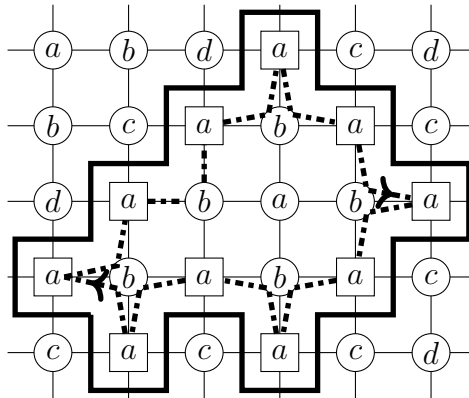


Figure 5.13: A region X delimited by the bold line and the only border B of X is represented by the square vertices. The dotted line represents the only walk in \mathcal{W}_B .

Proof. Suppose to the contrary that the configuration appears. Then, the cycle $xywz$ of length 4 is unbalanced. Thus, before switching, $xywz$ was already an unbalanced cycle of length 4 in (P, π) since balance is preserved by switching. By definition of (P, π) as a Cartesian product of cycles, the signs of xy and wz are the same. This is also the case for zx and wy . Thus this cycle is balanced (it has an even number of negative edges), a contradiction. \square

Definition 5.42. Two vertices x and y on the boundary of the region X are *border neighbors* if x and y have a common neighbor in X and a common neighbor in $P - X$ (see Figure 5.12). We note $BN(x)$ the set of border neighbors of x .

A *border* B of a region X is a subset of ∂X corresponding to an equivalence class for the transitive closure of the border neighborhood relation (see Figure 5.13). That is to say, two vertices x and y of ∂X are in the same border B of X if and only if there exists a sequence u_0, u_1, \dots, u_k of vertices of B such that $u_0 = x$, $u_k = y$ and for all $0 \leq i < k$, u_i and u_{i+1} are border neighbors.

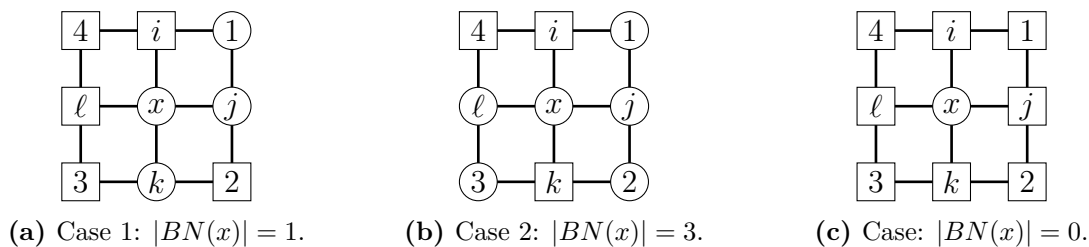


Figure 5.14: The two cases up to symmetry where $|BN(x)|$ has an odd number of vertices and the case where $BN(x) = \emptyset$. The square vertices represent vertices not in the region of x , the circular ones are in the region.

Claim 5.43. *All vertices of a border B of a region X have the same color called the color of B .*

Proof. By definition of B it suffices to show that any two border neighbors x and y have the same color. Let z be their common neighbor in X . Without loss of generality, suppose X is of type ab and z has color b . Since the coloring is proper, x and y have color a . \square

Claim 5.44. *A vertex x of a border B has an even number of border neighbors. Moreover if $BN(x) = \emptyset$, then $X = \{x\}$.*

Proof. If $|BN(x)|$ is odd, then we are in one of the first two cases of Figure 5.14. We will use the notation of the figure.

If $|BN(x)| = 1$, then up to rotation and symmetry, we can suppose that the vertex 1 is the border neighbor of x and that j is their common neighbor in X . Thus $i \notin X$. Now $\ell \notin X$, as otherwise the vertices $i, \ell, 4$ and x would be in the flat border configuration, which cannot be by Claim 5.41. The same argument implies $k \in X$ by considering x, k, j and 2. Thus x, ℓ, k and 3 are in the flat border configuration. A contradiction.

If $|BN(x)| = 3$, then up to rotation and symmetry, we can suppose that the vertex 4 is not a border neighbor of x . As 2 is a border neighbor of x , one of k and j is in X and the other is not. Without loss of generality, suppose $k \notin X$ and $j \in X$. As 3 is a border neighbor of x , we have $\ell \in X$. As 1 is a border neighbor of x , we have $i \notin X$. Thus 4, i, ℓ and x are in the flat border configuration, a contradiction.

Now if $BN(x) = \emptyset$, we can suppose that $i \notin X$ as x is in ∂X . Now to avoid the flat border configuration, j, k and ℓ must not be in X . This proves that $X = \{x\}$. \square

We can now define the set of walks associated with the border.

Definition 5.45. We associate with a border B of X , a set of closed walks \mathcal{W}_B in (P, π') included in X (see Figure 5.13). This set of walks delimits the border of X . We use $v_{i,j}$ to refer to the vertex $\overline{v_{i,j}}$ of P for concision.

We will define the walks piece by piece. In the particular case that X has only one vertex, then $\mathcal{W}_B = \emptyset$. Now we can suppose that for each $x \in B$, we have $BN(x) \neq \emptyset$ by Claim 5.44.

First pick an arbitrary vertex x of B . The vertex x is a border vertex thus there exists at least one vertex w adjacent to x which is not in X . In case there are more than one such vertex, we choose one of them arbitrarily. Up to rotation of the coordinate system, we can suppose $x = v_{i,j}$ and $w = v_{i,j+1}$. We will choose $y \in BN(x)$ according to the order in Figure 5.15(a). Meaning the first vertex among $v_{i+1,j+1}, v_{i+1,j-1}$ and $v_{i-1,j-1}$ that

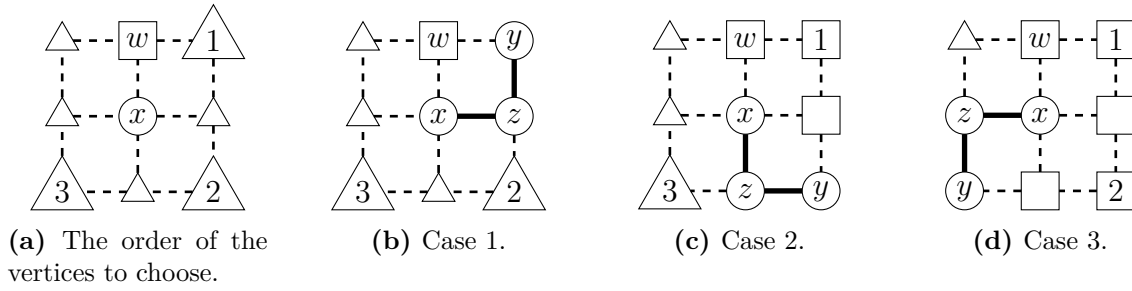


Figure 5.15: The first step in constructing the walk. The square vertices represent vertices not in the region of x , the circular ones are in the region while the triangular ones are undecided. The dashed edges are the edges of G^∞ while the bold edges are the edges of the walk.

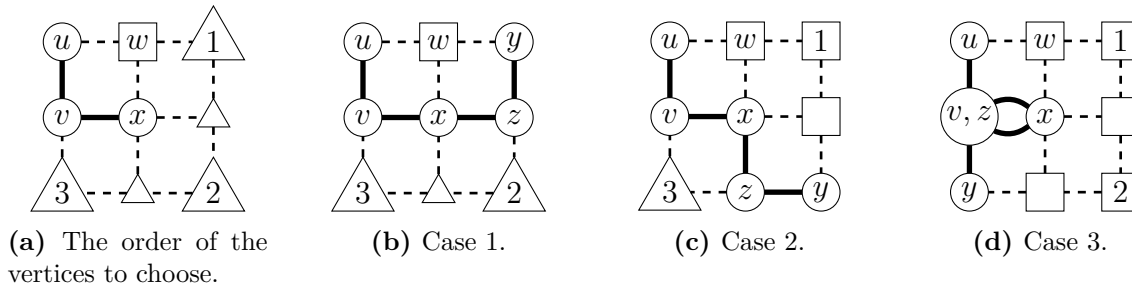


Figure 5.16: The next step in constructing the walk. We use the same notation as in Figure 5.15. We constructed $s_{l-2} = u$, $s_{l-1} = v$, $s_l = x$. We then construct $s_{l+1} = z$ and $s_{l+2} = y$.

belongs to $BN(x)$. The three cases are depicted in Figure 5.15(b), 5.15(c) and 5.15(d). Note that as $BN(x)$ is non-empty, $BN(x)$ has at least two vertices by Claim 5.44, thus we are in at least one of the three cases above.

Through the construction, the “turn left” property, which implies that the vectors $\overrightarrow{s_{2i}s_{2i+1}}$ and $\overrightarrow{s_{2i+1}s_{2i+2}}$ form a direct base, will be conserved. That is to say for every i , the vectors $\overrightarrow{s_{2i}s_{2i+1}}$ and $\overrightarrow{s_{2i+1}s_{2i+2}}$, which belong to $\{(1, 0), (-1, 0), (0, 1), (0, -1)\}$, are orthogonal and such that the angle between the two vectors is $\frac{\pi}{2}$ and not $-\frac{\pi}{2}$. Informally, if someone travels from s_{2i} to s_{2i+1} then this person would have to turn left to go to s_{2i+2} .

Now that we have x and y we can start to construct our walk W , by taking $s_0 = x$, $s_1 = z$ and $s_2 = y$ where z is the common neighbor of x and y which is in X . Note that the position of z is forced since otherwise we have a flat border configuration.

Suppose now that we have constructed the walk up to $s_0, \dots, s_{\ell-2}, s_{\ell-1}, s_\ell$ with ℓ even. If $s_{\ell-2} = s_0$, $s_{\ell-1} = s_1$ and $s_\ell = s_2$, then we stop and close this walk by removing the last two vertices. Otherwise we will construct $s_{\ell+1}$ and $s_{\ell+2}$. Suppose that $s_\ell = v_{i,j}$. Up to rotation of the coordinate system, we can suppose that $s_{\ell-2} = v_{i-1,j+1}$ and $s_{\ell-1} = v_{i-1,j}$. The vertex $s_{\ell-1}$ could in principle be $v_{i,j+1}$ but this would contradict the “turn left” property. We construct $s_{\ell+2}$ as the first vertex among $v_{i+1,j+1}$, $v_{i+1,j-1}$ and $v_{i-1,j-1}$ that belong to $BN(x)$ (see Figure 5.16(a)). The three cases are depicted in Figure 5.16(b), 5.16(c) and 5.16(d). As before, since $BN(x)$ is non-empty and of even cardinality, we are in one of those three cases. As in the first step, the vertex $s_{\ell+1}$ is the common neighbor of s_ℓ and $s_{\ell+2}$ in X .

If we stop and there are pairs of border neighbors that are not in the same walk, we can start the process again with this pair of vertices as the first and third vertices of the walk. See Figure 5.17 for an example where we need to construct another walk. To keep

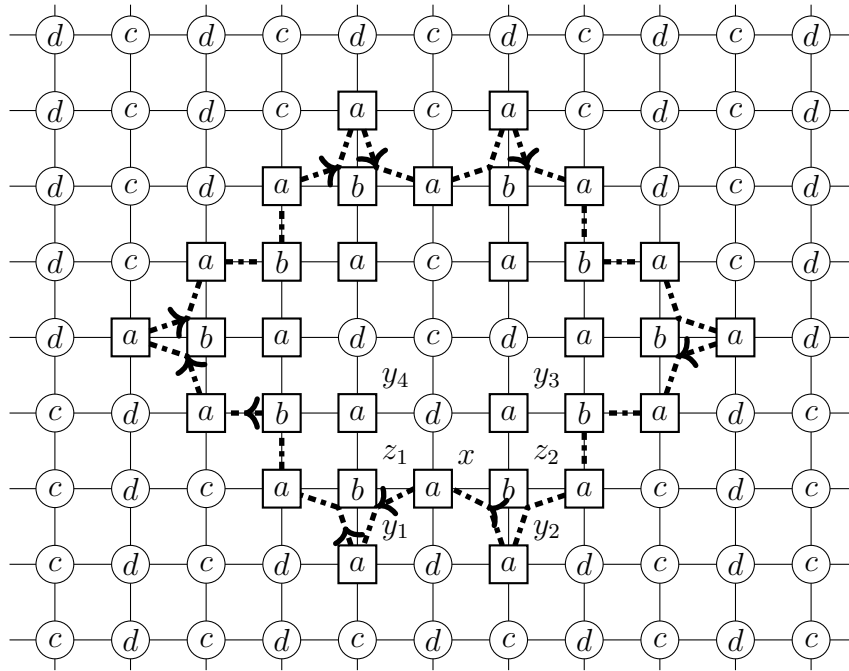


Figure 5.17: A closed walk W in \mathcal{W}_a where the vertex x of the walk does not have all of its border neighbors in W .

the assumption of the construction true, we must carefully choose the first vertex of the walk among these two in such a way that the “turn left” property is conserved. With the notation of Figure 5.17, the vertices x and y_4 are not in the same walk. To restart our process, we have two choices: $s_0 = x$, $s_1 = z_1$ and $s_2 = y_4$ or $s_0 = y_4$, $s_1 = z_1$ and $s_2 = x$. In the first case, the vectors $\overrightarrow{s_0 s_1}$ and $\overrightarrow{s_1 s_2}$, which are equal to $(-1, 0)$ and $(0, 1)$ respectively, form an indirect base of the plane. Hence we must choose $s_0 = y_4$, $s_1 = z_1$ and $s_2 = x$, which ensures that the “turn left” property holds.

Claim 5.46. *The construction of Definition 5.45 has the following properties:*

1. *the construction terminates and all walks are closed,*
2. *the walks are of even length,*
3. *all vertices with even indices have the same color,*
4. *all vertices with odd indices have the same color which is different from the color of the vertices with even indices,*
5. *all vertices of the border B are vertices of some walk with even index,*
6. *the number of occurrences of a vertex x of the border B in all the walks of \mathcal{W}_B is given by $|BN(x)|/2$.*

Proof. Suppose we do not terminate. As the number of possible edges is finite, the sequence we construct is ultimately periodic. Since s_0, s_1, s_2 do not appear consecutively in this order in the rest of the sequence, as we did not stop, the sequence is not periodic. Thus there exists a first moment at which there exist i and j such that $s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}$ and $s_{j-2}, s_{j-1}, s_j, s_{j+1}, s_{j+2}$ are subsequences of the sequence we constructed, verifying $s_{i-2} \neq s_{j-2}$, $s_{i-1} \neq s_{j-1}$, $s_i = s_j$, $s_{i+1} = s_{j+1}$ and $s_{i+2} = s_{j+2}$. Note that knowing s_{i-2} ,

s_i and s_{i+2} imposes the choices of s_{i-1} and s_{i+1} by the “turn left” property, this is why these two indices exist. Without loss of generality, up to rotating the grid, we can assume that $s_i = v_{x,y}$ and $s_{i+2} = v_{x+1,y+1}$. By reversing the construction, we can observe that s_{i-2} is the first border neighbor of s_i among $v_{x-1,y+1}$, $v_{x-1,y-1}$ and $v_{x+1,y-1}$ in this order. In this case, s_{j+2} and s_{i-2} are uniquely determined by construction and are equal, a contradiction. Now the process terminates, thus the walks are closed by definition of the terminating condition. This proves 1.

Since the walks are included in X which is bipartite, they have even length which proves 2. In a similar way, all vertices with even indices are on the border B of X thus they have the same color by Claim 5.43, thus 3 is true. Thus all vertices with odd indices have the other color of X which proves 4.

We already saw that the vertices of even indices are on B . Suppose that x is not part of a walk. We removed the case $BN(x) = \emptyset$ by not considering those B 's thus there exists $y \in BN(x)$. Then x and y are not in the same walk, thus we create a new one with these two vertices, a contradiction. This proves 5.

Similarly, if the number of occurrences is strictly smaller than $|BN(x)|/2$, we would have restarted the process in x . Now suppose that this number is strictly greater than $|BN(x)|/2$. Then there exists a pair of border neighbors x and y that belong to two walks (and there is a vertex z in between them in those two walks) by the pigeon hole principle. Since the construction of the walks only use the position of three consecutive vertices to decide the next two ones, the two walks are identical after passing through xzy . By construction, we can choose the first vertices of the walks arbitrarily among the vertices of even indices by shifting the indices, thus we can consider that the two walks start by xzy . Thus the two walks are identical which cannot be the case as we would not have restarted to create the second walk. \square

We define the set of closed walks \mathcal{W}_a as the union of all closed walks \mathcal{W}_B where B is a border with color a .

Take C to be a vertical or horizontal cycle of P . For the sake of simplicity, we will take C to be the vertical cycle induced by the vertices $\{\overline{v_{x_0,y}} : y \in \mathbb{Z}\}$ where $\ell = 2p + 1$ and $x_0 \in \{0, \dots, \ell - 1\}$ (i.e. C is a UC_q -layer). All the following definitions can be stated in the other case by symmetry.

Let W be a closed walk in P (resp. a representation of a closed walk of P in G^∞). We define a *positive crossing of C by W in P* (resp. G^∞) as a sub-walk $t_0, t_1, \dots, t_{k-1}, t_k$ of W (possibly going through the end of W and going back at the beginning) such that $t_0 = \overline{v_{x_0-1,y}}$ (resp. $t_0 = v_{x_0-1+n\ell,y}$ for $n \in \mathbb{Z}$) for some $y \in \mathbb{Z}$, the vertex t_i belongs to C (resp. is of the form $v_{x_0+n\ell,y'_i}$ for some $y'_i \in \mathbb{Z}$) for every $i \in \{1, \dots, k-1\}$ and $t_k = \overline{v_{x_0+1,y''}}$ (resp. $t_k = v_{x_0+1+n\ell,y''}$) for some $y'' \in \mathbb{Z}$. The set of positive crossings $Cross_P^+(W, C)$ (resp. $Cross_{G^\infty}^+(W, C)$) is the set of all-positive crossings of C by W in P (resp. G^∞).

We can similarly define a *negative crossing of C by W in P* (resp. G^∞) by a sub-walk $t_0, t_1, \dots, t_{k-1}, t_k$ of W (possibly going through the end of W and going back at the beginning) such that $t_0 = \overline{v_{x_0+1,y}}$ (resp. $t_0 = v_{x_0+1+n\ell,y}$ for $n \in \mathbb{Z}$) for some $y \in \mathbb{Z}$, the vertex t_i belongs to C (resp. is of the form $v_{x_0+n\ell,y'_i}$ for some $y'_i \in \mathbb{Z}$) for every $i \in \{1, \dots, k-1\}$ and $t_k = \overline{v_{x_0-1,y''}}$ (resp. $t_k = v_{x_0-1+n\ell,y''}$) for some $y'' \in \mathbb{Z}$. The set of negative crossings $Cross_P^-(W, C)$ (resp. $Cross_{G^\infty}^-(W, C)$) is the set of all-negative crossings of C by W in P (resp. G^∞).

Claim 5.47. *If W_{G^∞} represents W_P , then*

$$\left| \text{Cross}_P^+(W_P, C) \right| = \left| \text{Cross}_{G^\infty}^+(W_{G^\infty}, C) \right| \quad \text{and} \quad \left| \text{Cross}_P^-(W_P, C) \right| = \left| \text{Cross}_{G^\infty}^-(W_{G^\infty}, C) \right|.$$

Proof. We will only consider positive crossings, the proof for negative crossings is similar. By taking the quotient of a sub-walk of W_{G^∞} , we see that each crossing in G^∞ is also present in P . Thus $\left| \text{Cross}_P^+(W_P, C) \right| \geq \left| \text{Cross}_{G^\infty}^+(W_{G^\infty}, C) \right|$. Now take a crossing of C by W_P in P , it is a sub-walk of W_P . Thus if we take the corresponding sub-walk in W_{G^∞} , we get a crossing in G^∞ . Thus the two sets are equal. \square

One of our main results is the following proposition.

Proposition 5.48.

$$\sum_{W \in \mathcal{W}_a} \left(\left| \text{Cross}_P^+(W, C) \right| + \left| \text{Cross}_P^-(W, C) \right| \right) \equiv \{uv \in C : uv \text{ has color } ac \text{ or } ad\} \pmod{2}.$$

Proof. Take a vertex x of C in (P, π') colored a . If the region of x is $\{x\}$, then x has two incident edges colored ac or ad and x is not contained in any crossing as it does not belong to a walk in \mathcal{W}_a by definition. Thus we can ignore them.

If x has at least one incident edge colored ac or ad , then it belongs to some border colored a .

Now take a vertex x of color a in some walk $W \in \mathcal{W}_a$. Depending on the size of $BN(x)$ there are one or two occurrences of x in \mathcal{W}_a by Claim 5.46. Up to rotation we can suppose that we have $v_{i-1,j+1}, v_{i-1,j}, v_{i,j} = x$ as a sub-walk of W . Depending on the orientation of C (vertical or horizontal), for each sub-case, we must consider the two orientations. For one orientation there are four sub-cases: $|BN(x)| = 4$, $|BN(x)| = 2$ and $v_{i+1,j+1}$ is the border neighbor of x different from $v_{i-1,j+1}$, $|BN(x)| = 2$ and $v_{i+1,j-1}$ is the border neighbor of x different from $v_{i-1,j+1}$ or $|BN(x)| = 2$ and $v_{i-1,j-1}$ is the border neighbor of x different from $v_{i-1,j+1}$. For each case, the local structure of the walk is determined by our construction process and the “turn left” property. All the sub-cases are depicted in Figure 5.18. In each case the number of crossings for the sub-walks considered is equal, modulo 2, to the number of edges colored ab of x in C .

Now note that no vertices of color b in $W \in \mathcal{W}_a$ has both neighbors in the same layer. Thus a crossing of C by W always contains a vertex colored a of C . Thus all crossings are counted in the above case analysis.

Since for each edge colored ac or ad , the vertex colored a has a neighbor not in its region, it is on some border and thus we counted those edges in the case analysis or when we treated the case of the region of size one.

Thus the number of edges colored ac or ad in C is equal to the sum of the number of crossings of C by walks in \mathcal{W}_a modulo 2. \square

Claim 5.49. *The number of edges colored ac or ad in C is equal to the number of edges of C colored ab modulo 2.*

Proof. Let us call E_{ac} (resp. E_{ad} , resp. E_{ab}) the set of edges colored ac (resp. ad , resp. ab). Let $N'(x)$ be the set of edges of P incident with a vertex x . Since a vertex of color

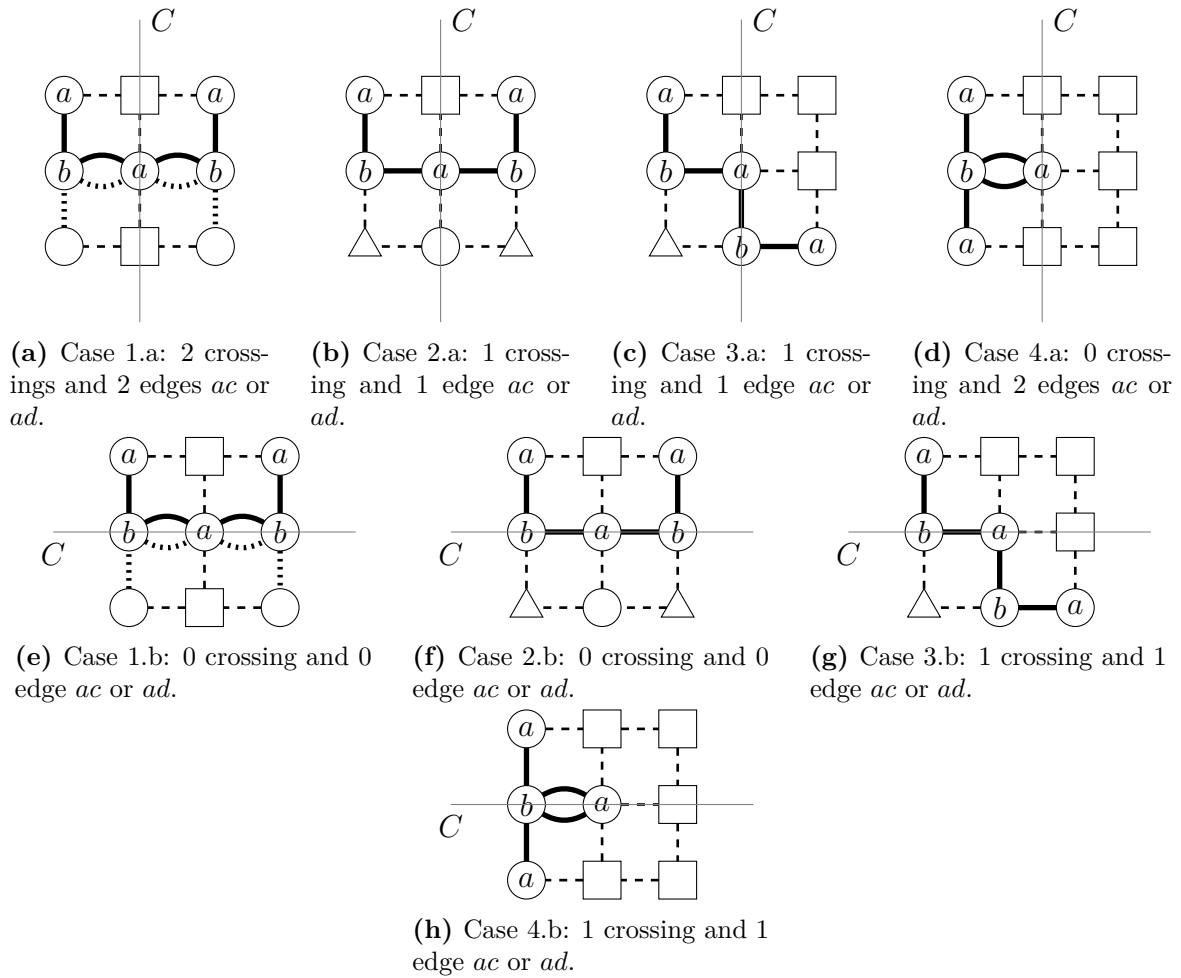


Figure 5.18: All cases for the central vertex to belong to a closed walk $W \in \mathcal{W}_a$. We use the same notation as in Figure 5.15. The dotted lines in sub-figures 5.18(a) and 5.18(e) are a second passage in the central vertex by a walk in \mathcal{W}_a (possibly the same as the bold line). The edges ac or ad are the edges between a circular vertex and a square vertex. For each case we count the number of crossings of the drawn walks and the number of edges of color ac or ad incident to the central vertex and belonging to the cycle C .

a has two incident edges in C , we have:

$$\begin{aligned}
 |(E_{ac} \cup E_{ad}) \cap E(C)| &\equiv \sum_{\substack{x \in C \\ \text{of color } a}} |(E_{ac} \cup E_{ad}) \cap N'(x) \cap E(C)| \pmod{2} \\
 &\equiv \sum_{\substack{x \in C \\ \text{of color } a}} \deg_C(x) - |E_{ab} \cap N'(x) \cap E(C)| \pmod{2} \\
 &\equiv \sum_{\substack{x \in C \\ \text{of color } a}} 2 - |E_{ab} \cap N'(x) \cap E(C)| \pmod{2} \\
 &\equiv \sum_{\substack{x \in C \\ \text{of color } a}} |E_{ab} \cap N'(x) \cap E(C)| \pmod{2} \\
 &\equiv |E_{ab} \cap E(C)| \pmod{2}.
 \end{aligned}$$

□

5.4.5 Crossings and turns

In this section, we will suppose that C is the vertical cycle of (P, π') equal to $\{\overline{v_{x_0, y}} : y \in \mathbb{Z}\}$ for $\ell = 2p + 1$ and $x_0 \in \{0, \dots, \ell - 1\}$. We identify the cycle C on (P, π') and the set $\{v_{x, y} : \overline{v_{x, y}} \in C\}$ of vertices of G^∞ . All what is defined below also works for a horizontal cycle with $\ell = q$. Here we want to connect the number of crossings of the previous section with the number of turns of Section 5.4.3.

Definition 5.50. Let $v_{x, y}$ be a vertex of G^∞ . We define the function g_C as follows:

$$g_C(v_{x, y}) = \left\lfloor \frac{x - x_0}{\ell} \right\rfloor.$$

For a (s_0, s_n) -walk W_{G^∞} in G^∞ , we define f_C as follows:

$$f_C(W_{G^\infty}) = g_C(s_n) - g_C(s_0).$$

Claim 5.51. For a $(v_{x, y}, v_{z, t})$ -walk W_{G^∞} of G^∞ representing a closed walk W_P of P such that $v_{x, y} \notin C$, we have:

$$f_C(W_{G^\infty}) \equiv \left| \text{Cross}_{G^\infty}^+(W_{G^\infty}, C) \right| - \left| \text{Cross}_{G^\infty}^-(W_{G^\infty}, C) \right| \pmod{2}.$$

Proof. Suppose $W_{G^\infty} = (s_i)_{i \in \{0, \dots, n\}}$, by assumption $s_0 \notin C$. This ensures that all crossings of C by W_{G^∞} are sub-walks that do not go through the end of W_{G^∞} and go back at the beginning. Take a crossing t_0, \dots, t_k . By definition of a crossing, the x coordinate of t_0 is of the form $x_0 + \varepsilon + n\ell$ for some $n \in \mathbb{Z}$ and $\varepsilon \in \{-1, 1\}$ and the x coordinate of t_k is of the form $x_0 - \varepsilon + n\ell$. Hence, we have $g_C(t_k) - g_C(t_0) = 1$ if the crossing is positive and $g_C(t_k) - g_C(t_0) = -1$ if it is negative.

Now we just have to show that the other sub-walks of W_{G^∞} do not contribute to $f_C(W_{G^\infty})$. We can write $W_{G^\infty} = W_0, W_0^{\text{cross}}, W_1, \dots, W_{k-1}^{\text{cross}}, W_k$ for some integer k where each W_i^{cross} is a crossing and the other sub-walks are not. Note that:

$$f_C(W_{G^\infty}) = \sum_{i \in \{0, \dots, k\}} f_C(W_i) + \sum_{i \in \{0, \dots, k-1\}} f_C(W_i^{\text{cross}}).$$

If for all $i \in \{0, \dots, k\}$, $f_C(W_i) = 0$, then $f_C(W_{G^\infty}) = \sum_{i \in \{0, \dots, k-1\}} f_C(W_i^{\text{cross}})$. As each positive crossing counts for 1 in the sum and each negative crossing counts for -1 in

the sum, $f_C(W_{G^\infty}) = |Cross_{G^\infty}^+(W_{G^\infty}, C)| - |Cross_{G^\infty}^-(W_{G^\infty}, C)|$ which implies our result when taking these numbers modulo 2.

Since the last vertices of the W_i 's are the same as the first vertices of the crossings, we know that they do not belong to C . The same is true for the first vertices of the W_i 's. Then, for a $(v_{x,y}, v_{z,t})$ -walk, we have $x, z \in \{x_0 + n\ell + 1, \dots, x_0 + n\ell + \ell - 1\}$ for some n . But in all cases the value of g_C is n . Indeed, if $x = x_0 + n\ell + \lambda$ with $1 \leq \lambda \leq \ell - 1$ then $g_C(v_{x,y}) = \lfloor \frac{x-x_0}{\ell} \rfloor = \lfloor \frac{n\ell + \lambda}{\ell} \rfloor = n + 0$. Thus $f_C(W_i) = 0$. This concludes the proof. \square

Claim 5.52. For a closed walk W_P in P and W_{G^∞} a representation of W_P on G^∞ :

$$f_C(W_{G^\infty}) \equiv \tau_x(W_P) \pmod{2}.$$

Proof. Suppose that W_{G^∞} is a $(v_{x,y}, v_{z,t})$ -walk. Note that $z = x + n\ell$ for some $n \in \mathbb{Z}$. We have:

$$\begin{aligned} \tau_x(W_P) &\equiv \left\lfloor \frac{z-x}{\ell} \right\rfloor \pmod{2} \\ &\equiv n \pmod{2}, \end{aligned}$$

while:

$$\begin{aligned} f_C(W_{G^\infty}) &\equiv \left\lfloor \frac{z-x_0}{\ell} \right\rfloor - \left\lfloor \frac{x-x_0}{\ell} \right\rfloor \pmod{2} \\ &\equiv n + \left\lfloor \frac{x-x_0}{\ell} \right\rfloor - \left\lfloor \frac{x-x_0}{\ell} \right\rfloor \pmod{2} \\ &\equiv n \pmod{2}. \end{aligned}$$

\square

5.4.6 End of the proof

We can now prove Lemma 5.30.

Proof. Note that by shifting the indices, we can suppose that the first vertex of each walk $W_P \in \mathcal{W}_a$ does not belong to C . By using Claim 5.49, Proposition 5.48, Claim 5.47, Claim 5.51 and Claim 5.52, in this order, we get:

$$\begin{aligned} |E_{ab} \cap E(C)| &\equiv \sum_{W_P \in \mathcal{W}_a} |Cross_P^+(W_P, C)| + |Cross_P^-(W_P, C)| \pmod{2} \\ &\equiv \sum_{\substack{W_P \in \mathcal{W}_a \\ W_{G^\infty} \text{ represents } W_P \text{ and} \\ \text{its first vertex } \notin C}} |Cross_{G^\infty}^+(W_{G^\infty}, C)| - |Cross_{G^\infty}^-(W_{G^\infty}, C)| \pmod{2} \\ &\equiv \sum_{\substack{W_P \in \mathcal{W}_a \\ W_{G^\infty} \text{ represents } W_P \text{ and} \\ \text{its first vertex } \notin C}} f_C(W_{G^\infty}) \pmod{2} \\ &\equiv \sum_{W_P \in \mathcal{W}_a} \tau_x(W_P) \pmod{2}. \end{aligned}$$

By the choice of C in the previous subsection, $C = UC_q$ and thus $|E_{ab} \cap C| \equiv 1 \pmod{2}$. Therefore:

$$1 \equiv \sum_{W_P \in \mathcal{W}_a} \tau_x(W_P) \pmod{2}.$$

By taking $C = BC_{2p+1}$, a horizontal cycle, we obtain:

$$0 \equiv |E_{ab} \cap E(C)| \equiv \sum_{W \in \mathcal{W}_a} \tau_y(W) \pmod{2}.$$

Recall that Proposition 5.39 states that:

$$0 \equiv q\tau_y(W) + \tau_x(W) \pmod{2}.$$

Thus:

$$0 \equiv q \times 0 + 1 \pmod{2}.$$

This is a contradiction. □

This concludes the proof of Lemma 5.30.

5.5 Perspectives

We studied the Cartesian product of signed graphs both on an algebraic point of view, and in relation with the chromatic number of signed graphs.

On top of the already mentioned open problems, it would be interesting to study the behavior of the Cartesian product with respect to other signed graphs parameters, the sign-preserving chromatic number being one of the more interesting of these.

Other interesting questions relate to other graph products. The tensor product of signed graphs \times was defined in [85] under the names of “strong product” and “categorical product”. For two signed graphs (G, σ) and (H, π) , the *tensor product* of (G, σ) and (H, π) , denoted $(G, \sigma) \times (H, \pi)$, is the signed graph with vertex set $V(G) \times V(H)$; $(G, \sigma) \times (H, \pi)$ contains the two edges $(u, w)(v, z)$ and $(v, w)(u, z)$ with sign $\sigma(uv)\pi(wz)$, for every edges $uv \in E(G)$ and $wz \in E(H)$.

One can also define the strong product of signed graphs as a generalization of the strong product of graphs [169]. For two signed graphs (G, σ) and (H, π) , the *strong product* of (G, σ) and (H, π) , denoted $(G, \sigma) \boxtimes (H, \pi)$, is the signed graph with vertex set $V(G) \times V(H)$, the set of positive edges (resp. negative edges) of $(G, \sigma) \boxtimes (H, \pi)$ is obtained from the union of the set of positive edges (resp. negative edges) of $(G, \sigma) \square (H, \pi)$ and the set of positive edges (resp. negative edges) of $(G, \sigma) \times (H, \pi)$.

For these products, the following are equivalent:

- (G, σ) and (H, π) are balanced,
- $(G, \sigma) \square (H, \pi)$ is balanced,
- $(G, \sigma) \times (H, \pi)$ is balanced, and
- $(G, \sigma) \boxtimes (H, \pi)$ is balanced.

In particular these products are all compatible with switching. We can therefore ask questions similar to the ones of this chapter for the tensor product and the strong product. Are there unique prime decompositions? Can we recognize them? What can we say about the chromatic number of a product?

Another product of signed graphs, called the lexicographical product of signed graphs, has been defined in [91] but unfortunately lacks these nice properties [31]. Finding a signature which would work well with switching for this product seems impossible. Under these conditions, this product behave more as a product of 2-edge-colored graphs than signed graphs.

Part II

Edge-coloring with constraints

Chapter 6

Introduction

In this second part, we study diverse notions of edge-coloring of undirected graphs. Contrary to signed graphs, here the colors of the edges are not part of the structure of the graph. The goal is as follows: we are given an undirected graph and we want to assign colors to its edges in such a way that some property holds.

The most well-known edge-coloring problem is the one requiring to find a proper edge-coloring with the least number of colors (see Section 1.2.4). In this setting, the constraints on the colors of the edges are completely characterized by the graph structure. It is easy to generate many other notions of edge-coloring by choosing a general rule which decides which edges can receive the same color and which cannot. This type of edge-coloring can be reduced to (vertex) coloring a graph which has $E(G)$ for vertex set. On that account, proper edge-coloring a graph G is equivalent to coloring the line graph $L(G)$ of G . In Section 6.1, we present a particular family of such edge-colorings where the constraints on the edges are determined by the distance separating them.

Another way to construct edge-coloring notions is to put constraints on subgraphs (possibly induced) composed of edges of the same color. For example, the *arboricity* of a graph G is the minimum number of colors required to edge-color G (not necessarily properly) in such a way that each subgraph composed of edges of the same color is a forest (see [155]).

Chapter 7 and Chapter 8 explore edge-coloring notions which are related to distance edge-coloring. In particular, both edge-coloring notions can be expressed as vertex coloring problems and can be reformulated as constraints on the subgraphs composed of edges of the same color.

Another family of edge-coloring problems is obtained by creating a vertex coloring from the edge-coloring of the graph. Here, given a graph G , the goal is to color the edges, with the fewest number of colors possible, in such a way that the vertex coloring obtained from the edge-coloring is proper. There are multiple ways to generate the vertex coloring from the edge-coloring, moreover we can also add constraints on the edge-coloring itself depending on the problem. We present some of these edge-coloring notions in Section 6.2.

Chapter 9 and Chapter 10 both explore such edge-coloring notions.

6.1 Distance edge-coloring

In 1983, Fouquet and Jolivet introduced the notion of strong edge-coloring. This coloring is a particular case of distance edge-coloring.

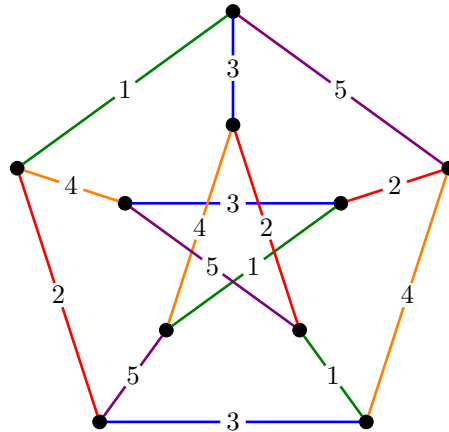


Figure 6.1: A strong 5-edge-coloring of the Petersen graph.

Definition 6.1 (*t*-distance edge-coloring). A k -edge-coloring c of a graph G is a *t*-distance k -edge-coloring if and only if for every two edges e and e' at distance at most t in G , $c(e) \neq c(e')$. The *t*-distance chromatic index of G , denoted $\chi'_t(G)$, is the smallest k for which G admits a *t*-distance k -edge-coloring.

A 2-distance edge-coloring is also called a *strong edge-coloring* (see [78]), and the 2-distance chromatic index of G is also called the *strong chromatic index* of G .

See Figure 6.1 for an example of a strong edge-coloring of the Petersen graph.

As mentioned before, *t*-distance edge-coloring of a graph G can be reduced to vertex coloring some graph which can be computed from G . This graph is a power graph of the line graph of G , *i.e.* the graph obtained from $L(G)$ by adding an edge between any two vertices which are at distance at most t . It is however often impractical to study the corresponding vertex coloring problem to obtain precise results on the *t*-distance edge-coloring. For more results on distance edge-coloring, we refer the reader to the PhD Thesis [58] and the articles [117, 122, 123, 181].

Strong edge-coloring has also been extensively studied, see for example [49, 64, 104, 106] or the PhD Thesis [100]. One particularly interesting result for us is the following upper bound for subcubic graphs.

Theorem 6.2 (Andersen [5] and Horák, Qing, and Trotter [107]). *For every subcubic graph G , we have $\chi'_2(G) \leq 10$.*

This bound is sharp: a $K_{3,3}$ with one subdivided edge cannot be strong edge-colored with 9 colors (see figure 6.2).

6.2 Distinguishing vertices with edge-colors

An interesting family of edge-coloring problems is the concept of *distinguishing labellings*, where the aim, given an undirected graph, is to edge-color the graph so that adjacent vertices get distinguished by some function f computed from the edge-colors. Note that, in this context, the edge-colors are often called *labels* and we talk about *k*-labellings instead of *k*-edge-colorings. As reported in a survey [81] by Gallian on the topic, there actually exist dozens and dozens different types of distinguishing labelling notions, which all have their own particular behaviors and subtleties.

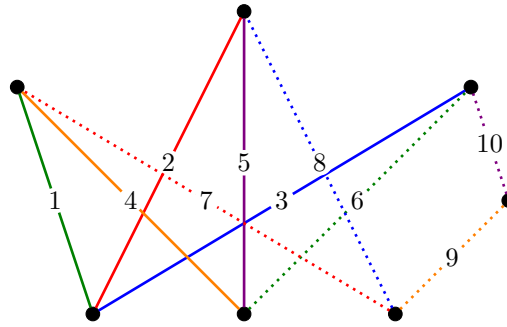


Figure 6.2: A strong 10-edge-coloring of $K_{3,3}$ with one subdivided edge.

In this thesis, we mainly study two different notions of distinguishing labellings. For both, the function f , which determines the color of the vertex, associates with a vertex u a color obtained from the labels (*i.e.* edge-colors) of the edges incident with u . To simplify notation, we often say that a label c is *incident with* a vertex v if there is an edge e labelled c incident with v .

The weakest labelling notion which satisfies this scheme is the one for which the multi-set of labels incident with a vertex u is the vertex color of u . For such a labelling, two adjacent vertices cannot receive the same multi-sets of labels. One can see that a graph containing an isolated edge cannot be labelled with this definition: the two endpoints of the edge would receive the same vertex color and would not be distinguished. Graphs without connected components isomorphic to K_2 are called *nice graphs*. In [186], Vučković showed that every nice graph G admits a 3-labelling (*i.e.* 3-edge-coloring) for which adjacent vertices have different multi-sets of incident labels (see Chapter 10 for more details).

Stronger variants can be obtained in two ways. The first way to obtain more restrictive edge-coloring notions is to add constraints to the edge-coloring: an *adjacent vertex distinguishing coloring* of a graph is a *proper* edge-coloring of the graph which distinguishes adjacent vertices through their sets of incident edge-colors. Note that the literature on adjacent vertex distinguishing coloring does not use the term “labelling” for edge-colors. We study this variant in Chapter 9.

An other way to obtain other distinguishing labellings is to use more restrictive functions to compute the vertex color. For example, if the labels are integers, one can sum or multiply them to obtain the vertex color. The famous 1-2-3-Conjecture [125] states that every nice graph G admits a 3-labelling for which the vertex coloring, obtained by summing the labels on edges incident with the colored vertex, is proper. We prove the product version of this conjecture (where we multiply labels instead of summing them) in Chapter 10.

Chapter 7

Between proper and strong edge-colorings of subcubic graphs

Due to a remarkable result of Vizing [184] (see Theorem 1.49), we know that the minimum number of colors needed to color the edges of a graph G , the chromatic index $\chi'(G)$ of G , is either $\Delta(G)$ or $\Delta(G) + 1$. Recall that the graphs with the former value of the chromatic index are commonly said to be class I, and the other ones class II.

In this chapter, we are interested in subcubic multi-graphs. Through this chapter, we use the term graph to designate multi-graphs (possibly with loops). We need at most 4 colors to color such graphs, the complete graph on four vertices with one edge subdivided being the smallest representative of a class II subcubic graph, and the Petersen graph being the smallest 2-connected class II cubic graph. For subcubic graphs of class II, it has been shown that they can be colored in such a way that one of the colors (usually denoted δ) is used relatively rarely (cf. [3, 79]). This motivates the question whether the edges of color δ can be pairwise distant. Payan [161] and independently Fouquet and Vanherpe [79] proved that every subcubic graph with chromatic index 4 admits a proper edge-coloring such that the edges of one color are at distance at least 3 from each other, *i.e.*, the end-vertices of those edges induce a matching in the graph.

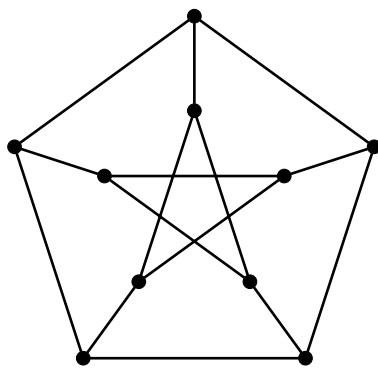
Gastineau and Togni [84] investigated a generalization of edge-colorings taking into account the distance between edges of the same color.

Definition 7.1 (*S*-packing). For a given non-decreasing sequence of integers $S = (s_1, \dots, s_k)$, an *S*-packing edge-coloring of a graph G is a decomposition of $E(G)$ into disjoint sets X_1, \dots, X_k , where the edges in the set X_i are pairwise at distance at least $s_i + 1$.

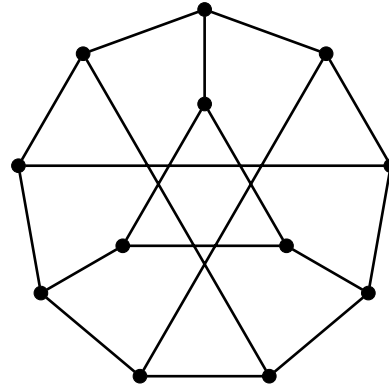
A set X_i is called an *s_i-packing*; a 1-packing is simply a matching, and a 2-packing is an induced matching.

To simplify the notation, we denote repetitions of same elements in S using exponents, e.g., $(1, 2, 2, 2)$ can be written as $(1, 2^3)$.

The notion of an *S*-packing edge-coloring is motivated by its vertex counterpart, introduced by Goddard and Xu [87] as a natural generalization of the packing chromatic number [86]. In [84], the authors consider *S*-packing edge-coloring of subcubic graphs with prescribed number of 1's in the sequence. Vizing's result translated to *S*-packing edge-coloring gives that every subcubic graph admits a $(1, 1, 1, 1)$ -packing edge-coloring, while class I subcubic graphs are $(1, 1, 1)$ -packing edge-colorable. Moreover, by Payan's, Fouquet's and Vanherpe's result, every subcubic graph admits a $(1, 1, 1, 2)$ -packing edge-coloring.



(a) The Petersen graph.



(b) The Tietze graph.

Figure 7.1: The Petersen graph (a) and the Tietze graph (b) admit a $(1, 1, 1, 2)$ -packing edge-coloring, and 2 cannot be increased to 3.

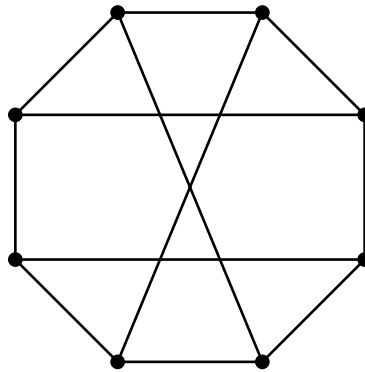


Figure 7.2: The Wagner graph is the smallest cubic graph which needs 10 colors for a strong edge-coloring.

Theorem 7.2 (Payan [161], and Fouquet and Vanherpe [79]). *Every subcubic graph admits a $(1, 1, 1, 2)$ -packing edge-coloring.*

Here 2 cannot be changed to 3, due to the Petersen graph and the Tietze graph (depicted in Figure 7.1). They both have chromatic index 4, and we need at least two edges of each color. Since every two edges are at distance at most 3, we have the tightness. However, Gastineau and Togni do believe that the following conjecture is true.

Conjecture 7.3 (Gastineau and Togni [84]). *Every cubic graph different from the Petersen graph and the Tietze graph is $(1, 1, 1, 3)$ -packing edge-colorable.*

Clearly, reducing the number of 1's in sequences increases the total number of needed colors, *i.e.*, the length of the sequence. In fact, if there is no 1 in a sequence, then the coloring is a strong edge-coloring. Theorem 6.2, which states that every subcubic graph admits a strong edge-coloring with at most 10 colors, can be reformulated using packing edge-coloring: every subcubic graph admits a (2^{10}) -packing edge-coloring. Another example of tightness of this result is given by the Wagner graph in Figure 7.2 which needs 10 colors for a strong edge-coloring. Let us remark here that the Wagner graph is class I, meaning that smallest chromatic index does not necessarily mean less number of colors for a strong edge-coloring of a graph.

Proper and strong edge-coloring of subcubic graphs have been studied extensively already in the previous decades. More recently, Gastineau and Togni [84] started filling



Figure 7.3: The smallest subcubic graph which does not admit a $(1, 1, 2^3)$ -packing edge-coloring (Conjecture 7.4.1) nor a $(1, 2^6)$ -packing edge-coloring (Conjecture 7.4.2), and the smallest class I subcubic graph which does not admit a $(1, 1, 2^2)$ -packing edge-coloring (Conjecture 7.4.3) nor a $(1, 2^5)$ -packing edge-coloring (Conjecture 7.4.4).

the gap by considering $(1^k, 2^\ell)$ -packing edge-colorings for $k \in \{1, 2\}$. They proved that every cubic graph with a 2-factor admits a $(1, 1, 2^5)$ -packing edge-coloring, and the number of required 2-packings reduces by one if the graph is class I. For the case with one 1-packing, they remark that using the bound for the strong edge-coloring one obtains that every subcubic graph admits a $(1, 2^9)$ -packing edge-coloring. These bounds are clearly not tight, and they propose a conjecture (the items 1 and 3 in Conjecture 7.4), which motivated the research presented in this chapter. The case 2 has been formulated as a question, and we added the case 4, due to affirmative results of computer tests on subcubic graphs of small orders.

Conjecture 7.4. *Every subcubic graph G admits:*

1. a $(1, 1, 2^4)$ -packing edge-coloring [84],
2. a $(1, 2^7)$ -packing edge-coloring [84],
3. a $(1, 1, 2^3)$ -packing edge-coloring if G is class I [84],
4. a $(1, 2^6)$ -packing edge-coloring if G is class I.

The conjectured bounds, if true, are tight. For Conjecture 7.4.1 and Conjecture 7.4.2, a subcubic graph that achieves the upper bound is the complete bipartite graph $K_{3,3}$ with one subdivided edge (the left graph in Figure 7.3). Recall that this graph is also class II and needs 10 colors for a strong edge-coloring, hence achieving the upper bounds for proper edge-coloring, strong edge-coloring, $(1, 1, 2^4)$ -packing edge-coloring, and $(1, 2^7)$ -packing edge-coloring. Indeed, each 1-packing contains at most three edges and each 2-packing contains at most one edge. An analogous argument holds for Conjecture 7.4.3 and Conjecture 7.4.4 on the complete bipartite graph $K_{3,3}$.

Conjecture 7.4 bridges two of the most important edge-colorings, proper and strong, basically claiming that each 1-packing could be replaced by three 2-packings. Indeed, if such operations were possible, then one could transform a $(1, 1, 1, 2)$ -packing edge-coloring into a strong edge-coloring. Note that this does not apply to subclasses of graphs, e.g., the Wagner graph needs 10 colors for a strong edge-coloring and it is class I.

This chapter contributes to answering the conjecture by providing upper bounds with one additional color for all four cases of Conjecture 7.4.

Theorem 7.5. *Every subcubic graph G admits:*

1. a $(1, 1, 2^5)$ -packing edge-coloring,
2. a $(1, 2^8)$ -packing edge-coloring,
3. a $(1, 1, 2^4)$ -packing edge-coloring if G is class I,
4. a $(1, 2^7)$ -packing edge-coloring if G is class I.

This work is joint work with Hervé Hocquard and Borut Lužar. An extended abstract of this chapter is published in the proceedings of the international conference IWOCA 2020 [101]. A full version is also available on arXiv [102].

This research has been done in the scope of the bilateral project between France and Slovenia, BI-FR/19-20-PROTEUS-001. The third author of the corresponding paper was partly supported by the Slovenian Research Agency Program P1-0383 and the project J1-1692.

The structure of the chapter is the following. We begin by presenting notation, definitions and auxiliary results in Section 7.1. In Section 7.2, we give proofs of the cases 1 and 3 of Theorem 7.5. In Sections 7.3 and 7.4, we prove the cases 2 and 4 of Theorem 7.5 in even stronger settings. We conclude the chapter with an overview of open problems and possible further work on this topic in Section 7.5.

Contents

7.1 Preliminaries	178
7.2 Proofs of Theorem 7.5.1 and Theorem 7.5.3	179
7.3 Proof of Theorem 7.5.2	180
7.4 Proof of Theorem 7.5.4	184
7.4.1 Auxiliary results	184
7.4.2 Proof	192
7.5 Further Work	200
7.5.1 Planar graphs	200
7.5.2 Bipartite graphs	202
7.5.3 Graphs with large girth	202

7.1 Preliminaries

For a vertex v , we denote the set of edges incident to v by $N'(v)$, and the edges incident to the neighbors of v (including the edges in $N'(v)$) by $N''(v)$. We refer to the former as the *edge-neighborhood of v* and to the latter as the *2-edge-neighborhood of v* . Analogously, we define the edge-neighborhood and the 2-edge-neighborhood of an edge e .

When coloring the edges, we deal with two types of colors. The ones allowing the edges of those colors to be at distance 2 from each other are called the *1-colors*, and the one requiring the edges to be at distance at least 3 from each other are called the *2-colors*. An edge colored with a 1-color (resp. a 2-color) is a *1-edge* (resp. a *2-edge*). We denote the number of 1-edges (resp. 2-edges) incident with a vertex v by $d_1(v)$ (resp. $d_2(v)$). For an edge uv , we denote by $A_2(uv)$ the number of available 2-colors, *i.e.*, the 2-colors with which the edge can be colored without violating the coloring requirements.

Sometimes, we will need a more careful analysis for choosing colors from the lists of available colors. For that purpose, we will use the classical result due to Hall [90].

Theorem 7.6 (Hall's Theorem [90]). *Let $\mathcal{A} = (A_i)_{i \in I}$ be a finite family of (not necessarily distinct) subsets of a finite set A . A system of representatives for the family \mathcal{A} is a set $\{a_i, i \in I\}$ of distinct elements of A such that $a_i \in A_i$ for all $i \in I$. \mathcal{A} has a system of representatives if and only if $|\bigcup_{i \in J} A_i| \geq |J|$ for all subsets J of I .*

Perhaps the strongest tool for determining if one can always choose colors from the lists of available colors such that given conditions are satisfied is the Combinatorial Nullstellensatz (see Section 1.4.4).

Theorem 1.77 (Combinatorial Nullstellensatz [4]). *Let \mathbb{F} be an arbitrary field, and let $P = P(X_1, \dots, X_n)$ be a polynomial in $\mathbb{F}[X_1, \dots, X_p]$. Suppose that the coefficient of a monomial $X_1^{k_1} \dots X_p^{k_p}$, where each k_i is a non-negative integer, is non-zero in P and the degree $\deg(P)$ of P equals $\sum_{i=1}^p k_i$.*

If moreover S_1, \dots, S_p are any subsets of \mathbb{F} with $|S_i| > k_i$ for $i = 1, \dots, p$, then there are $s_1 \in S_1, \dots, s_p \in S_p$ so that $P(s_1, \dots, s_p) \neq 0$.

When considering lists of available colors for an edge, we are in fact dealing with the list version of the packing edge-coloring. In each use case, we actually apply the Combinatorial Nullstellensatz with 2-colors, thus limiting ourselves to the list version of strong edge-coloring. For any list edge-assignment L , if a graph G admits a strong edge-coloring σ such that $\sigma(e) \in L(e)$ for all edges in $E(G)$, then we say that G is *strong L -edge-choosable* or σ is a *strong L -edge-coloring* of G . The graph G is *strong k -edge-choosable* if it is strong L -edge-choosable for every k -list edge-assignment L . The *strong edge-choosability* $ch_2'(G)$ of G is the minimum k such that G is strong k -edge-choosable.

We will use the following result, due to Horňák and Woźniak [200], which deals with adjacent vertex-distinguishing list edge-coloring of cycles, *i.e.* proper list edge-coloring where the sets of colors for every pair of adjacent vertices are distinct (see Chapter 9). It is easy to see that such a coloring is also a strong edge-coloring of a cycle, and we write the statement in this language.

Theorem 7.7 (Horňák and Woźniak [200]). *Let n be an integer with $n \geq 3$. We have:*

1. $ch_2'(C_n) = 5$ if $n = 5$,
2. $ch_2'(C_n) = 4$ if $n \not\equiv 0 \pmod{3}$,
3. $ch_2'(C_n) = 3$ if $n \equiv 0 \pmod{3}$.

7.2 Proofs of Theorem 7.5.1 and Theorem 7.5.3

We begin with the cases of Theorem 7.5 using two 1-colors. These two cases provide straightforward extensions of the results by Gastineau and Togni [84], who established them for bridgeless cubic graphs. The extension comes from the following easy observation.

Proposition 7.8. *Let G be a connected subcubic graph and let X be a set of edges in G such that every two edges in X are at distance exactly 2 from each other. Then, X contains at most five edges. Moreover, if $|X| = 5$, then G is a cubic graph with 10 vertices.*

Proof. Let X be a set of edges in a subcubic graph G satisfying assumptions of the proposition. Each edge $e \in X$, has at most four adjacent edges, say e_1, e_2, e_3 , and e_4 . Each edge $e_i, 1 \leq i \leq 4$, can be adjacent to at most one other edge from X , since otherwise there would be two edges at distance 1 in X . This means, X contains at most five edges.

In the case where $|X| = 5$, every edge of G not in X and adjacent to an edge of X , connects two edges of X , hence every vertex of G is an end-vertex of some edge from X and thus the number of vertices in G is 10. Since every edge from X is adjacent to four edges, we infer that G is cubic. \square

Now, we are ready to prove Theorem 7.5.1 and Theorem 7.5.3.

Proof of Theorem 7.5.1 and Theorem 7.5.3. We begin with Theorem 7.5.1. Let G be a subcubic graph (we may assume it is connected) and let π be a $(1, 1, 1, 2)$ -packing edge-coloring of G (which exists by Theorem 7.2). To establish the statement, we only need to replace one 1-color in π with four 2-colors. Let X be the set of all the edges in G colored by one 1-color in π , and let G^* be the graph obtained from G by contracting all the edges in X (and removing loops that are created in the process). Clearly, G^* has maximum degree at most 4, and it is 4-colorable by Brooks' Theorem, unless it is isomorphic to K_5 . Observe that the vertex coloring of G^* induces a strong edge-coloring of the edges in X . Furthermore, by Proposition 7.8, the only graphs in which it may happen that five colors are needed to color G^* , are cubic with 10 vertices. For these graphs we have determined (using a computer) that they admit a $(1, 1, 2^4)$ -packing edge-coloring. This establishes Theorem 7.5.1.

Theorem 7.5.3 follows immediately from the argument above, since we do not have an extra 2-color in the coloring π . \square

7.3 Proof of Theorem 7.5.2

In order to prove Theorem 7.5.2, we prove a bit stronger result. We say that a $(1, 2^8)$ -packing edge-coloring of a subcubic graph G with the color set $\{0, 1, \dots, 8\}$, where 0 is a 1-color and the others are 2-colors, is a *good* $(1, 2^8)$ -packing edge-coloring if no 2^- -vertex of G is incident with a 1-edge (*i.e.* an edge colored 0).

Theorem 7.9. *Every subcubic graph admits a good $(1, 2^8)$ -packing edge-coloring.*

Proof. We prove Theorem 7.9 by contradiction. Let G be a minimal counter-example to the theorem in terms of $|V(G)| + |E(G)|$. Clearly, G is connected and has maximum degree 3. In the following claims, we establish some structural properties of G which will eventually yield a contradiction with the existence of G . In most of the claims, we consider a graph G' smaller than G , which, by minimality of G , admits a good $(1, 2^8)$ -packing edge-coloring π , and we show that π can be extended to G by recoloring some edges of G' and coloring the edges of G not being colored by π .

We start by proving that G is a simple 2-connected cubic graph.

Claim 7.10. *G is simple.*

Proof. Suppose there are vertices u and v in G connected by at least two parallel edges. Let e be one of these edges. Remove e from G to obtain a smaller graph G' , and let π be a good $(1, 2^8)$ -packing edge-coloring of G' . We can extend π to G , since $A_2(e) \geq 1$, and hence there is an available 2-color for e .

Loops can be removed in a similar fashion. ■

Claim 7.11. *G is cubic.*

Proof. Suppose first that there exists a 1-vertex v adjacent to a vertex u in G . By minimality of G , there exists a good $(1, 2^8)$ -packing edge-coloring π of $G' = G - v$, meaning u is not incident with a 1-edge. We can extend π to G , by coloring uv with any of the (at least two) 2-colors that do not appear in the 2-edge-neighborhood of u . Hence, G does not contain 1-vertices.

Suppose now that there exists a 2-vertex v adjacent to two vertices u and w . By minimality of G , $G' = G - v$ admits a good $(1, 2^8)$ -packing edge-coloring π , and hence u and w are not incident with a 1-edge. We show that π can be extended to G as follows. First observe that if $A_2(uv) \geq 2$ and $A_2(vw) \geq 1$, or $A_2(uv) \geq 1$ and $A_2(vw) \geq 2$, then π can be extended and we are done. So, we may assume $A_2(uv) \leq 1$ and $A_2(vw) \leq 1$. It follows that u and w are both 3-vertices in G , and moreover, u and w are not adjacent. Next, let u_1 and u_2 be the neighbors of u distinct from v , and analogously, let w_1 and w_2 be the neighbors of w . By the above argument, $d_2(u_1) + d_2(u_2) \geq 5$, meaning that at least one of u_1 and u_2 is a 3-vertex not incident with a 1-edge, say u_1 . Moreover, u_1 is not adjacent to w . Now, we recolor uu_1 with 0. By an analogous argument, we can recolor one of the edges adjacent to w with 0, obtaining a contradiction on the number of available colors. Hence, π can be extended to G . ■

Claim 7.12. *G is 2-connected.*

Proof. Since G is cubic, the claim is equivalent to saying that G is bridgeless. Suppose the contrary and let uv be a bridge in G . Let G_u (resp. G_v) be the component of $G - uv$ containing u (resp. v). By minimality of G , there is a good $(1, 2^8)$ -packing edge-coloring π_u of $G_u + uv$ and a good $(1, 2^8)$ -packing edge-coloring π_v of $G_v + uv$. The edge uv is in both cases colored with a 2-color. Now, we permute (if necessary) the 2-colors in the coloring π_v so that the color of uv is the same in both colorings, π_u and π_v , and the colors on the other two edges incident with v are distinct from the colors on the other two edges incident with u (except possibly the color 0). In this way, we obtain a good $(1, 2^8)$ -packing edge-coloring of G , a contradiction. ■

From now on, we show that G has girth greater than any constant $k \in \mathbb{N}$. Hence, we show that G is a 2-connected cubic tree, a contradiction.

We start by showing that the graph has no triangles and no 4-cycles.

Claim 7.13. *G does not contain triangles.*

Proof. Suppose the contrary and let $C = uvw$ be a triangle in G . If C is adjacent to two other triangles, then G is the complete graph on four vertices, and hence $(1, 2^8)$ -packing edge-colorable. So, we may assume C is adjacent to at most one triangle. We consider two cases.

Suppose first that C is adjacent to a triangle $C' = uvx$. Let w' and x' be the neighbors of w and x , respectively, distinct from u and v . Since G is bridgeless, by Claim 7.12, $w' \neq x'$. Now, by minimality of G , there is a good $(1, 2^8)$ -packing edge-coloring π of $G - \{uv, vw, wx, xu\} + w'x'$ (if $w'x'$ are already connected, we add a parallel edge). Let $\pi(w'x') = a$ and consider the coloring of G induced by π by coloring xx' and ww' by a . The edges uw , ux , vw , and vx have each at least five available 2-colors, while the edge uv has at least seven available 2-colors. This means that we are always able to extend π to G . Thus, C is not adjacent to any triangle.

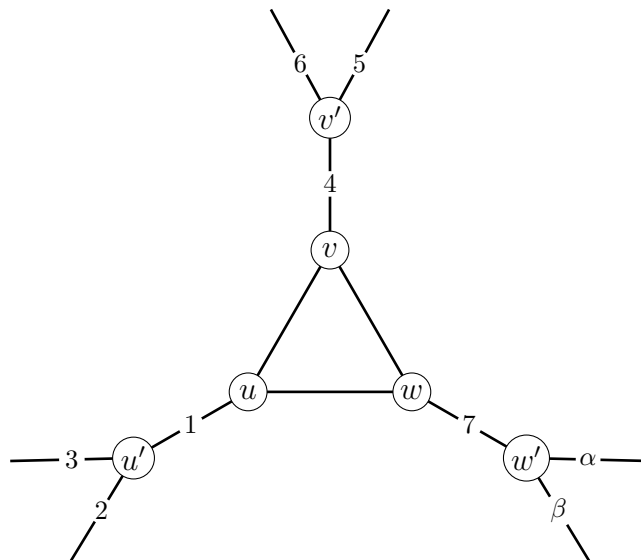


Figure 7.4: In a triangle where all three pendent edges are colored with a 2-color, we cannot forbid the same 2-color on all three edges.

Hence, the third neighbors of u , v , and w (denote them u' , v' , and w' , resp.), are all distinct. Let G' be the graph obtained from G by removing C and adding a new vertex x adjacent to the vertices u' , v' , and w' . Let π be a good $(1, 2^8)$ -packing edge-coloring of G' and let $a = \pi(u'x)$, $b = \pi(v'x)$, and $c = \pi(w'x)$. Let φ be the partial coloring of G induced by π , and set $\varphi(u'u) = a$, $\varphi(v'v) = b$, and $\varphi(w'w) = c$. Notice that only the edges uv , uw , and vw are not colored yet in φ , and each of them has at most seven colored neighbors in its 2-edge-neighborhood. If $0 \in \{a, b, c\}$, say $a = 0$, then for each edge of C there are two available 2-colors, and moreover, the edge vw can be colored with 0, so φ can be extended to all the edges.

Hence, we may assume a , b , and c are all 2-colors and moreover, they all have the same one 2-color available, otherwise φ can be extended to all the edges using the color 0 and two of the available 2-colors. This means that every non-colored edge must see the same seven colors in its colored 2-edge-neighborhood. Suppose uv has 7 distinct colors in its 2-edge-neighborhood (as depicted in Figure 7.4). Then, in order to have the same forbidden colors for uw and vw , it must hold that $\{\alpha, \beta\} = \{5, 6\}$ and $\{\alpha, \beta\} = \{2, 3\}$, respectively. We obtain a contradiction and so φ can always be extended to all the edges of G . ■

Claim 7.14. G does not contain 4-cycles.

Proof. We again proceed by contradiction. Suppose there is a 4-cycle $C = uvwz$ in G . Let u' , v' , w' , and z' be the neighbors of u , v , w , and z , respectively which do not belong to C . Since G has girth at least 4, the eight edges uv , vw , wz , zu , uu' , vv' , ww' , and zz' are distinct. Note that it is possible for the vertices u' and w' (resp. v' and z') to be equal; in such a case, there is at least one more 2-color available for the uncolored edges at distance two from this vertex. This counter-balance the fact that we may need to use one more 2-color on the edges incident with this vertex. Therefore, we can assume that u' , v' , w' and z' are distinct. We construct the graph $G' = (G \setminus V(C)) \cup \{u'w', v'z'\}$ (see Figure 7.5(b)). By minimality of G , G' admits a good $(1, 2^8)$ -packing edge-coloring π , and we show that we can always extend π to all the edges of G . We consider three cases

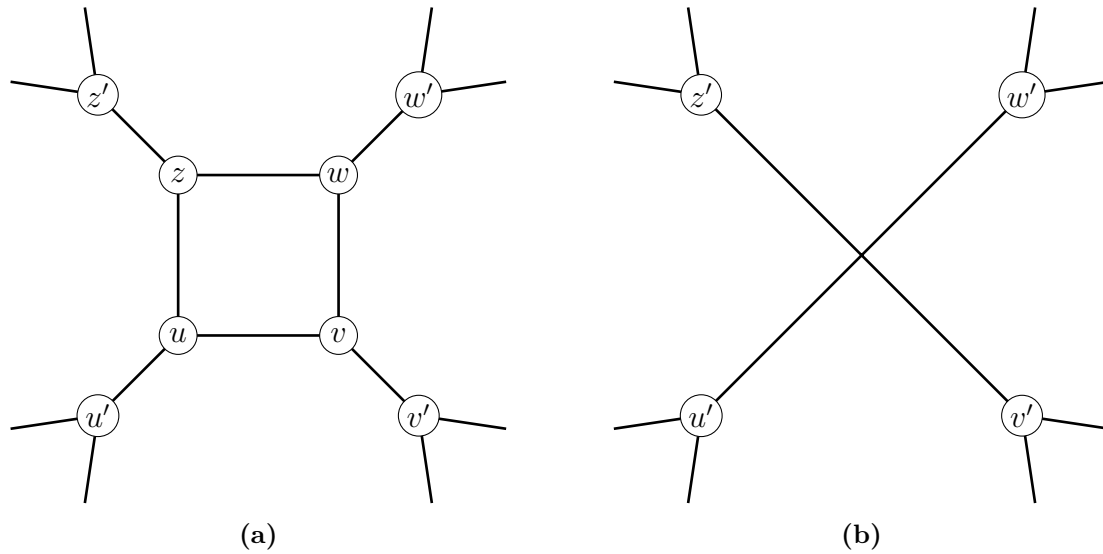


Figure 7.5: A 4-cycle with its neighborhood in G (a), and its replacement in the graph G' (b).

regarding the colors of the edges $u'w'$ and $v'z'$ in π .

- Both $u'w'$ and $v'z'$ are colored with the 1-color 0. Then the coloring φ of G induced by π has only the edges of C non-colored, while the four pendent edges of C (the edges with one end-vertex in C) are colored with 0. This means that there are at least four available 2-colors for every edge of C , so we can complete the coloring by Theorem 7.6.
- One of the edges, say $u'w'$, is colored with 0, and the others with a 2-color, say 1. Then, in the induced coloring φ , we have $\varphi(u'u) = \varphi(w'w) = 0$ and $\varphi(v'v) = \varphi(z'z) = 1$. Now, the color 1 appears on two edges of every 2-edge-neighborhood of the edges of C . So, each of them has at least three available 2-colors. We consider two subcases. If the union of all sets of available colors contains at least four distinct colors, we can always choose distinct colors for all the edges, by Hall's theorem.

So, we may assume that all four edges have the same set of three available colors, say $\{6, 7, 8\}$. This means that on the edges incident to u' and v' there are colors 2, 3, 4, and 5. The same four colors must appear on the edges incident to v' and w' , but this implies that at least two edges of the same 2-color are at distance 2 in G' , a contradiction.

- Both, $u'w'$ and $v'z'$, are colored with some 2-color. In this case, we may color two opposite edges of C with the color 0. The remaining two non-colored edges have at least two available colors each, so we can always complete the coloring. This completes the proof of the claim. ■

We can now forbid cycles of greater length.

Claim 7.15. G contains no cycle of length at least 5.

Proof. Suppose the contrary and let $C = u_1u_2 \dots u_n$ be a minimal induced n -cycle in G , with $n \geq 5$. For every i , $1 \leq i \leq n$, let u'_i be the neighbor of the vertex u_i not in C , and let $G' = G \setminus V(C)$. Note that the u'_i are pairwise distinct by minimality of C . Then, by

minimality of G , there is a good $(1, 2^8)$ -packing edge-coloring π of G' . Since π is good, no u'_i is incident with the color 0. So, in the coloring φ of G induced by π , we can color every edge $u_i u'_i$ with 0. In this way, only the edges of C are left non-colored and each edge of C has at least four 2-colors available.

Suppose first that $n = 5$. We can color C , except if all five edges have the same four 2-colors available. If we are in this case, then suppose that 1 and 2 are the two colors on the edges incident to u'_1 , and 3 and 4 are the two colors on the edges incident to u'_2 . Then $\{1, 2\}$ must also be on the edges incident to u'_3 , $\{3, 4\}$ on the edges incident to u'_4 , and again $\{1, 2\}$ on the edges incident to u'_5 . Thus the edge $u_1 u_5$ has five available 2-colors, a contradiction.

If $n \geq 6$, then we can complete the coloring by Theorem 7.7, a contradiction. \blacksquare

By Claims 7.11-7.15, G is a cubic bridgeless graph without cycles, a contradiction. This concludes the proof of Theorem 7.9. \square

7.4 Proof of Theorem 7.5.4

We split this section into two parts. First, we introduce notation and auxiliary results, and then use them to prove Theorem 7.5.4 in a stronger setting.

7.4.1 Auxiliary results

To show that certain graphs are strongly edge-colorable from given lists, we will use the Combinatorial Nullstellensatz, *i.e.* Theorem 1.77. For this purpose we introduce the following. For two positive integers k and ℓ , where $k \leq \ell$, we define the polynomial $P_{k,\ell}$ as follows:

$$P_{k,\ell}(X_k, \dots, X_\ell) = (X_{k+1} - X_k) \cdot \prod_{i=k+2}^{\ell} (X_i - X_{i-2})(X_i - X_{i-1}). \quad (7.1)$$

If $k = \ell$, by convention we take $P_{k,\ell}(X_k) = 1$. Furthermore, for a monomial m , we denote by $p_{k,\ell}(m)$ the coefficient of m in the polynomial $P_{k,\ell}$.

Proposition 7.16. *For $k + 2 \leq \ell$, we have the following equalities:*

$$p_{k,\ell} \left(X_k \left(\prod_{i=k+1}^{\ell-2} X_i^2 \right) X_{\ell-1} X_\ell \right) = \begin{cases} -1 & \text{if } \ell - k \equiv 0 \pmod{3}, \\ 1 & \text{if } \ell - k \equiv 1 \pmod{3}, \\ 0 & \text{if } \ell - k \equiv 2 \pmod{3}, \end{cases} \quad (7.2)$$

$$p_{k,\ell} \left(X_k \left(\prod_{i=k+1}^{\ell-1} X_i^2 \right) \right) = \begin{cases} 0 & \text{if } \ell - k \equiv 0 \pmod{3}, \\ -1 & \text{if } \ell - k \equiv 1 \pmod{3}, \\ 1 & \text{if } \ell - k \equiv 2 \pmod{3}, \end{cases} \quad (7.3)$$

$$p_{k,\ell} \left(X_k \left(\prod_{i=k+1}^{\ell-2} X_i^2 \right) X_{\ell-1} X_\ell \right) = - p_{k,\ell} \left(X_k X_{k+1} \left(\prod_{i=k+2}^{\ell-1} X_i^2 \right) X_\ell \right), \quad (7.4)$$

$$p_{k,\ell} \left(X_k \left(\prod_{i=k+1}^{\ell-1} X_i^2 \right) \right) = - p_{k,\ell} \left(\left(\prod_{i=k+1}^{\ell-1} X_i^2 \right) X_\ell \right). \quad (7.5)$$

Proof. First, note that by shifting the indices we can assume, without loss of generality, that $k = 1$. Next, let $a_\ell = p_{1,\ell}(X_1 X_2^2 \dots X_{\ell-2}^2 X_{\ell-1} X_\ell)$ and $b_\ell = p_{1,\ell}(X_1 X_2^2 \dots X_{\ell-2}^2 X_{\ell-1}^2)$. By expanding the factor $(X_\ell - X_{\ell-2})(X_\ell - X_{\ell-1})$ of $P_{1,\ell}$, we obtain the following equalities on a_ℓ and b_ℓ for $\ell \geq 3$:

$$\begin{aligned} a_\ell &= -p_{1,\ell-1}(X_1 X_2^2 \dots X_{\ell-3}^2 X_{\ell-2} X_{\ell-1}) - p_{1,\ell-1}(X_1 X_2^2 \dots X_{\ell-3}^2 X_{\ell-2}^2) \\ &= -a_{\ell-1} - b_{\ell-1}, \\ b_\ell &= p_{1,\ell-1}(X_1 X_2^2 \dots X_{\ell-3}^2 X_{\ell-2} X_{\ell-1}) \\ &= a_{\ell-1}. \end{aligned}$$

Thus,

$$a_\ell = -a_{\ell-1} - a_{\ell-2}.$$

Moreover, $a_1 = 1$ and $a_2 = 0$, thus $a_3 = -1$. By induction, we infer Equalities (7.2) and (7.3) for a_ℓ and b_ℓ .

Symmetrically, by expanding the factor $(X_3 - X_1)(X_2 - X_1)$ of $P_{1,\ell}$, we infer analogous recurrences, and consequently Equalities (7.4) and (7.5) follow. We omit the proof. \square

A graph with k distinct edges e_1, \dots, e_k is an (a_1, a_2, \dots, a_k) -graph if its i -th edge e_i is associated with a list of colors L_i of size at least a_i . We say that an (a_1, a_2, \dots, a_k) -graph is *strongly choosable* (or (a_1, a_2, \dots, a_k) -choosable) if it admits a strong edge-coloring verifying that the color of e_i belongs to L_i for every assignment of L_i 's. A (a_1, a_2, \dots, a_k) -path is an (a_1, a_2, \dots, a_k) -graph which is a path.

To simplify the notation, we denote repetitions of same elements using exponents e.g., a $(2, 3, 3, 3, 1)$ -path is abbreviated as a $(2, 3^3, 1)$ -path.

Now, we show strong choosability of several configurations that will be used later in the proof.

Lemma 7.17. *For any positive integer ℓ , $\ell \geq 3$, a path of length ℓ is*

1. $(2, 2, 3^{\ell-3}, 2)$ -choosable if $\ell \not\equiv 0 \pmod{3}$,
2. $(2, 3^{\ell-2}, 2)$ -choosable.

Proof. Let P be an ℓ -path with the consecutive edges e_1, \dots, e_ℓ , each edge e_i having a list of available colors L_i for every i , $1 \leq i \leq \ell$. Moreover, to each edge e_i , $1 \leq i \leq \ell$, we associate the variable X_i .

Consider the first case. By Theorem 1.77, if the coefficient of $X_1 X_2 \left(\prod_{i=3}^{\ell-1} X_i^2 \right) X_\ell$ is non-zero, then there is a solution $(x_1, \dots, x_\ell) \in L_1 \times \dots \times L_\ell$ such that $P_{1,\ell}(x_1, \dots, x_\ell) \neq 0$. By Equation (7.2) of Proposition 7.16, this coefficient is non-zero if and only if $\ell - 1 \not\equiv 2 \pmod{3}$, thus only in the case when ℓ is not a multiple of 3. This proves the first case.

Now, we proceed with the second case. If P is a $(2, 3^{\ell-2}, 2)$ -path, then it is also a $(2, 2, 3^{\ell-3}, 2)$ -path. Thus, by the first case, it suffices to consider the case where ℓ is a multiple of 3. By Equation (7.3) of Proposition 7.16, the coefficient of $X_1 \left(\prod_{i=2}^{\ell-1} X_i^2 \right)$ is 1 if $\ell \equiv 0 \pmod{3}$, and so, by Theorem 1.77, P is strongly edge-colorable from its lists. This completes the proof. \square

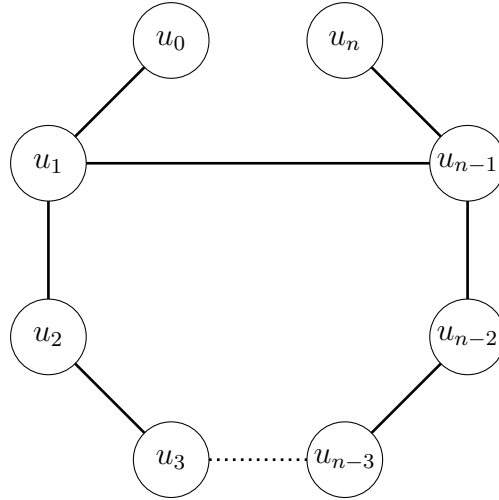


Figure 7.6: The graph D_n . Note that we do not color the edge $u_1 u_{n-1}$ when considering this configuration.

Note that in the second case of Lemma 7.17 we proved a stronger result; namely, a $(2, 3^{\ell-2}, 1)$ -path is strongly choosable if $\ell \equiv 0 \pmod{3}$.

Proposition 7.18. *For a positive integer ℓ , let X be a set of colors with $|X| = 3$, and P be a $(2, 3, \dots, 3, a)$ -path of length $3\ell + 1$, with $a \in \{2, 3\}$, such that $L_i \subseteq X$ for every $i \in \{1, \dots, 3\ell + 1\}$. If σ is a strong edge-coloring of P with $\sigma(e_i) \in L_i$, then $\sigma(e_1) = \sigma(e_{3\ell+1}) \in L_1 \cap L_{3\ell+1}$.*

Proof. Without loss of generality, let $X = \{1, 2, 3\}$, $L_0 = \{1, 2\}$, and $\sigma(e_1) = 1$. Then, $\sigma(e_2), \sigma(e_3) \in \{2, 3\}$, and hence $\sigma(e_4) = 1$. By induction, $\sigma(e_{3j+1}) = 1$ and $\sigma(e_{3j-1}), \sigma(e_{3j}) \in \{2, 3\}$. Thus, $\sigma(e_{3\ell+1}) = 1 = \sigma(e_1)$. \square

Given an n -path $P = u_0 u_1 \dots u_n$, we define the graph D_n as the graph obtained from P by adding the edge $u_1 u_{n-1}$ (see Figure 7.6). In D_n , we will only color the edges of the initial path but with restrictions established also by the edge $u_1 u_{n-1}$.

Now, we define polynomials that will be used in proving the strong choosability of D_n . For an integer n , $n \geq 5$, we define the following:

$$C_n(X_1, \dots, X_n) = (X_1 - X_{n-1})(X_1 - X_n)(X_2 - X_n)(X_2 - X_1) \prod_{i=3}^n ((X_i - X_{i-2})(X_i - X_{i-1})),$$

and

$$D_n(X_1, \dots, X_n) = C_n \cdot (X_2 - X_{n-1}).$$

Note that they correspond to the coloring polynomial of the edges of C_n and D_n . For example, if (x_1, \dots, x_n) is a non-zero solution of C_n , then it corresponds to a valid strong edge-coloring of the cycle of length n where $c(e_i) = x_i$. For a monomial m , by $c_n(m)$ we denote the coefficient of m in C_n . Similarly, $d_n(m)$ is the coefficient of m in D_n .

Lemma 7.19. *For every integer n , $n \geq 10$, D_n is $(3, 4, 2, 3, 5, 3, 2, 3^{n-9}, 4, 2)$ -choosable.*

Proof. By Theorem 1.77, it suffices to show

$$d_n(X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^3 X_n) \neq 0.$$

By the definition of D_n , we have

$$\begin{aligned} & d_n(X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^3 X_n) \\ &= c_n(X_1^2 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^3 X_n) - c_n(X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^2 X_n). \end{aligned}$$

Let

$$\alpha = c_n(X_1^2 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^3 X_n)$$

and

$$\beta = -c_n(X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^2 X_n).$$

We consider the values of α and β separately.

Claim 7.20. *For every integer $n \geq 10$, we have $\alpha = 0$.*

Proof. Suppose to the contrary that there is some n for which $\alpha \neq 0$. Then by Theorem 1.77, a $(3, 3, 2, 3, 5, 3, 2, 3^{n-9}, 4, 2)$ -cycle is strongly choosable. To reach a contradiction, we will find a $(3, 3, 2, 3, 5, 3, 2, 3^{n-9}, 4, 2)$ -cycle, for every n , which is not strongly choosable.

We distinguish three cases depending on the value of $n \pmod{3}$.

- *Suppose $n \equiv 0 \pmod{3}$.* Set $L_1 = \{1, 2, 3\}$, $L_2 = \{1, 2, 3\}$, $L_3 = \{2, 3\}$, $L_4 = \{1, 2, 3\}$, $L_5 = \{1, 2, 3, 4, 5\}$, $L_6 = \{1, 3, 4\}$, $L_7 = \{1, 4\}$, $L_8 = \{1, 3, 4\}, \dots, L_{n-3} = \{1, 3, 4\}$, $L_{n-2} = \{2, 3, 4\}$, $L_{n-1} = \{1, 2, 3, 4\}$, and $L_n = \{1, 2\}$ (see Figure 7.7(a)).

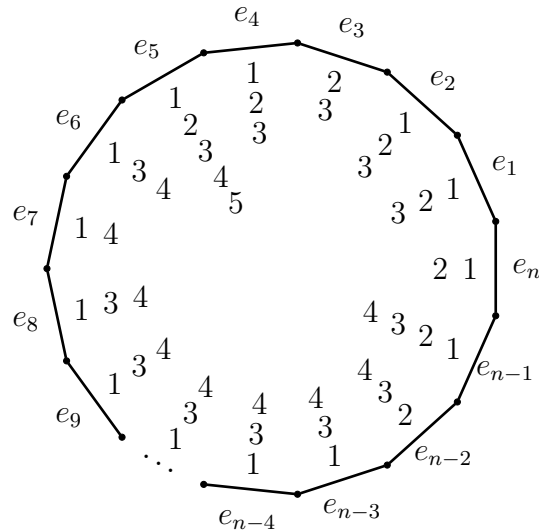
By Proposition 7.18, using the path e_3, e_2, e_1, e_n , we have $\sigma(e_n) = \sigma(e_3) = 2$. Then, the color 2 is forbidden in L_{n-2} , so we may assume $L_{n-2} = \{3, 4\}$. By using Proposition 7.18 on the path $e_7, e_8, \dots, e_{n-3}, e_{n-2}$, we infer $\sigma(e_{n-2}) = \sigma(e_7) = 4$. Now, since there are only colors 1 and 3 to color e_1, e_2 , and e_4 , by symmetry, we may set $\sigma(e_1) = \sigma(e_4) = 1$ and $\sigma(e_2) = 3$. This forces $\sigma(e_{n-1}) = 3$. Now the colors for the edges e_{n-3}, \dots, e_8 , and e_6 are forced and $\sigma(e_6) = 1$. But, e_4 and e_6 are at distance 2, so the cycle cannot be colored.

- *Suppose $n \equiv 1 \pmod{3}$.* Set $L_1 = \{1, 2, 3\}$, $L_2 = \{1, 2, 3\}$, $L_3 = \{2, 3\}$, $L_4 = \{2, 3, 5\}$, $L_5 = \{1, 2, 3, 4, 5\}$, $L_6 = \{1, 4, 5\}$, $L_7 = \{1, 4\}$, $L_8 = \{1, 3, 4\}, \dots, L_{n-3} = \{1, 3, 4\}$, $L_{n-2} = \{1, 2, 4\}$, $L_{n-1} = \{1, 2, 3, 4\}$, and $L_n = \{1, 2\}$ (see Figure 7.7(b)).

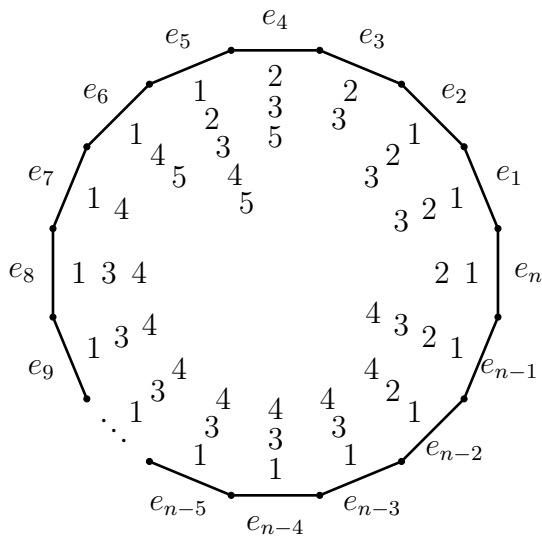
Again, using the path e_3, e_2, e_1, e_n , by Proposition 7.18, we have $\sigma(e_n) = \sigma(e_3) = 2$. Similarly, using the path e_7, e_8, \dots, e_{n-3} , we infer $\sigma(e_{n-3}) \in \{1, 4\}$. Now, $\sigma(e_{n-2})$ and $\sigma(e_{n-3})$ both belong to $\{1, 4\}$ and thus $\sigma(e_{n-1}) = 3$. Hence, we deduce the following colors: $\sigma(e_1) = 1$, $\sigma(e_2) = 3$, $\sigma(e_4) = 5$, $\sigma(e_{n-4}) = 3, \dots, \sigma(e_9) = 3$, $\sigma(e_7), \sigma(e_8) \in \{1, 4\}$, and $\sigma(e_6) = 5$. But, e_4 and e_6 are at distance 2, and so the cycle cannot be colored.

- *Suppose $n \equiv 2 \pmod{3}$.* Set $L_1 = \{1, 2, 3\}$, $L_2 = \{1, 2, 3\}$, $L_3 = \{2, 3\}$, $L_4 = \{1, 2, 3\}$, $L_5 = \{1, 2, 3, 4, 5\}$, $L_6 = \{1, 3, 4\}$, $L_7 = \{1, 3\}$, $L_8 = \{1, 3, 4\}, \dots, L_{n-4} = \{1, 3, 4\}$, $L_{n-2} = \{1, 3, 4\}$, $L_{n-2} = \{1, 2, 3\}$, $L_{n-1} = \{1, 2, 3, 4\}$, and $L_n = \{1, 2\}$ (see Figure 7.7(c)).

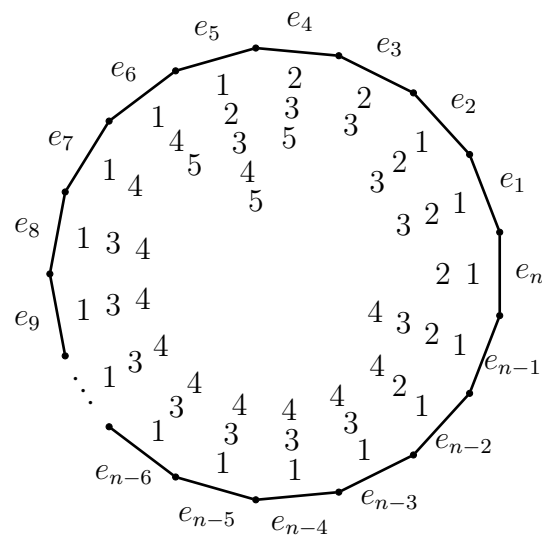
Using the path e_3, e_2, e_1, e_n , by Proposition 7.18, we again have $\sigma(e_n) = \sigma(e_3) = 2$, and hence $\sigma(e_{n-2}) \in \{1, 3\}$. Then, using the path e_7, e_8, \dots, e_{n-4} , we infer $\sigma(e_{n-4}) \in \{1, 3\}$. Now, $\sigma(e_{n-2})$ and $\sigma(e_{n-4})$ both belong to $\{1, 3\}$ thus $\sigma(e_{n-3}) = 4$. After coloring these three edges, we can assume $\sigma(e_{n-1}) = 1$. It follows that we have $\sigma(e_1) = 3$, $\sigma(e_2) = 1$, $\sigma(e_4) = 3$, $\sigma(e_{n-2}) = 3$, $\sigma(e_{n-4}) = 1, \dots, \sigma(e_9) = 3$, $\sigma(e_8) = 4$,



(a) Case: $n \equiv 0 \pmod{3}$.



(b) Case: $n \equiv 1 \pmod{3}$.



(c) Case: $n \equiv 2 \pmod{3}$.

Figure 7.7: Three cases for cycles of length n which are not $(3, 3, 2, 3, 5, 3, 2, 3^{n-9}, 4, 2)$ -choosable depending on the value of $n \pmod{3}$. For each cycle, the edge e_i is given the list corresponding to the numbers present between the edge and the ‘center’ of the cycle. For example, in Figure 7.7(a), the edge e_4 is given the list $\{1, 2, 3\}$ while the edge e_5 is given the list $\{1, 2, 3, 4, 5\}$. These lists can also be found in the proof of Claim 7.20.

$\sigma(e_7) = 1$, and $\sigma(e_6) = 3$. But e_4 and e_6 are at distance 2, and so the cycle cannot be colored.

Since there are non-strongly edge-colorable cycles for every length n , we obtain a contradiction. Thus $\alpha = 0$. \blacksquare

Claim 7.21. *For every integer $n \geq 12$, we have $\beta \neq 0$.*

Proof. We first define the polynomial Q_n as

$$Q_n = (X_{n-1} - X_{n-2})(X_{n-1} - X_{n-3})(X_n - X_{n-1})(X_n - X_{n-2}) \\ (X_1 - X_{n-1})(X_1 - X_n)(X_2 - X_n).$$

Observe that $C_n = Q_n \times P_{1,n-2}$. Since $P_{1,n-2}$ does not contain X_{n-1} and X_n , it suffices to find the coefficients of Q_n having $X_{n-1}^2 X_n$ as a factor to calculate β . Let us write $Q_n = \sum_{i,j} X_{n-1}^i X_n^j R_{n,i,j}$, where $R_{n,i,j}$ is a polynomial in X_1, \dots, X_{n-2} . We have

$$R_{n,2,1} = X_1^2 X_2 X_{n-3} + X_1^2 X_{n-3} X_{n-2} + X_1 X_2 X_{n-3} X_{n-2} \\ + X_1^2 X_{n-2}^2 + X_1 X_{n-3} X_{n-2}^2 + X_2 X_{n-3} X_{n-2}^2.$$

For a monomial m , we denote by $r_{n,2,1}(m)$ the coefficient of m in $R_{n,2,1}(m)$. So, we may write β as

$$\begin{aligned} \beta &= -c_n (X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2}^2 X_{n-1}^2 X_n) \\ &= -r_{n,2,1}(X_1^2 X_2 X_{n-3}) \cdot \overbrace{p_{1,n-2}(X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3} X_{n-2}^2)}^{\beta_1} \\ &\quad - r_{n,2,1}(X_1^2 X_{n-3} X_{n-2}) \cdot \overbrace{p_{1,n-2}(X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3} X_{n-2})}^{\beta_2} \\ &\quad - r_{n,2,1}(X_1 X_2 X_{n-3} X_{n-2}) \cdot \overbrace{p_{1,n-2}(X_1 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3} X_{n-2})}^{\beta_3} \\ &\quad - r_{n,2,1}(X_1^2 X_{n-2}^2) \cdot \overbrace{p_{1,n-2}(X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3})}^{\beta_4} \\ &\quad - r_{n,2,1}(X_1 X_{n-3} X_{n-2}^2) \cdot \overbrace{p_{1,n-2}(X_1 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3})}^{\beta_5} \\ &\quad - r_{n,2,1}(X_2 X_{n-3} X_{n-2}^2) \cdot \overbrace{p_{1,n-2}(X_1^2 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4} X_{n-3})}^{\beta_6} \\ &= -(\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6). \end{aligned} \tag{7.6}$$

It remains to determine the coefficients of $P_{1,n-2}$ for the monomials appearing in Equation (7.6). We compute them by reducing them in simpler forms. In particular, we make use of the facts that

$$P_{k,\ell}(X_k, \dots, X_\ell) = (X_{k+1} - X_k)(X_{k+2} - X_k) \cdot P_{k+1,\ell}(X_{k+1}, \dots, X_\ell),$$

$$P_{k,\ell}(X_k, \dots, X_\ell) = (X_\ell - X_{\ell-1})(X_\ell - X_{\ell-2}) \cdot P_{k,\ell-1}(X_k, \dots, X_{\ell-1}).$$

and that X_5 must appear in four terms of $P_{1,n-2}$, meaning that X_5 must be chosen in each of these terms when we expand the polynomial. The same is true for X_1 when it is raised to the power 3 as it appears in three terms.

Let us now compute the six coefficients.

$$\begin{aligned}
\beta_1 &= p_{1,n-2}(X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= p_{2,n-2}(X_2 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= -p_{3,n-2}(X_4 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= -p_{4,n-2}(X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= -p_{5,n-2}(X_5^2 X_6 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= -p_{6,n-2}(X_6 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}^2) \\
&= -p_{6,n-3}(X_6 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
&= \begin{cases} -1 & \text{if } n \equiv 0 \pmod{3}, \\ 1 & \text{if } n \equiv 1 \pmod{3}, \\ 0 & \text{if } n \equiv 2 \pmod{3}. \end{cases} \quad \text{by Proposition 7.16: Equations (7.4) and (7.2)}
\end{aligned}$$

$$\begin{aligned}
\beta_2 &= p_{1,n-2}(X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= p_{2,n-2}(X_2^2 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= p_{3,n-2}(X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= p_{4,n-2}(X_4 X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -p_{5,n-2}(X_5^2 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -p_{6,n-2}(X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -p_{7,n-2}(X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= \begin{cases} 1 & \text{if } n \equiv 0 \pmod{3}, \\ -1 & \text{if } n \equiv 1 \pmod{3}, \\ 0 & \text{if } n \equiv 2 \pmod{3}. \end{cases} \quad \text{by Proposition 7.16: Equation (7.2)}
\end{aligned}$$

$$\begin{aligned}
\beta_3 &= p_{1,n-2}(X_1 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -p_{2,n-2}(X_2^2 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&\quad - p_{2,n-2}(X_2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -\beta_2 + p_{3,n-2}(X_3 X_4 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&\quad + p_{3,n-2}(X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= -\beta_2 - p_{4,n-2}(X_4 X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) + \beta_2 \\
&= p_{5,n-2}(X_5^2 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= p_{6,n-2}(X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= p_{7,n-2}(X_7 X_8^2 \dots X_{n-4}^2 X_{n-3} X_{n-2}) \\
&= \begin{cases} -1 & \text{if } n \equiv 0 \pmod{3}, \\ 1 & \text{if } n \equiv 1 \pmod{3}, \\ 0 & \text{if } n \equiv 2 \pmod{3}. \end{cases} \quad \text{by Proposition 7.16: Equation (7.2)}
\end{aligned}$$

Note that when reducing $p_{4,n-2}$, we used the fact that X_5^4 cannot appear in any monomial of $P_{4,n-2}$. Similarly, X_k has at most power 2 in the monomials of $P_{k,\ell}$.

$$\begin{aligned}
 \beta_4 &= p_{1,n-2}(X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= p_{2,n-2}(X_2^2 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= p_{3,n-2}(X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= p_{4,n-2}(X_4 X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= -p_{5,n-2}(X_5^2 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= -p_{6,n-2}(X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= -p_{7,n-2}(X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}^2) \\
 &= -p_{7,n-3}(X_7 X_8^2 \dots X_{n-5}^2 X_{n-4} X_{n-3}) \\
 &= \begin{cases} 0 & \text{if } n \equiv 0 \pmod{3}, \\ 1 & \text{if } n \equiv 1 \pmod{3}, \\ -1 & \text{if } n \equiv 2 \pmod{3}. \end{cases} \quad \text{by Proposition 7.16: Equation (7.2)}
 \end{aligned}$$

$$\begin{aligned}
 \beta_5 &= p_{1,n-2}(X_1 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= -p_{2,n-2}(X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= -p_{3,n-2}(X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= -p_{4,n-2}(X_4 X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= -p_{5,n-2}(X_5^3 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= 0 \quad \text{as there is no monomial with } X_5^3 \text{ in } P_{5,n-2}.
 \end{aligned}$$

$$\begin{aligned}
 \beta_6 &= p_{1,n-2}(X_1^2 X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= p_{2,n-2}(X_2^2 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= p_{3,n-2}(X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-4}^2 X_{n-3}) \\
 &= -\beta_5 \\
 &= 0.
 \end{aligned}$$

Hence, inserting the values in Equation (7.6), we obtain

$$\beta = \begin{cases} 1 & \text{if } n \equiv 0 \pmod{3}, \\ -2 & \text{if } n \equiv 1 \pmod{3}, \\ 1 & \text{if } n \equiv 2 \pmod{3}. \end{cases}$$

■

From Claims 7.20 and 7.21, we have

$$d_n(X_1^2 X_2^3 X_3 X_4^2 X_5^4 X_6^2 X_7 X_8^2 \dots X_{n-2} X_{n-1}^3 X_n) = \alpha - \beta \neq 0.$$

This establishes the lemma. □

For smaller values of n , we prove another result.

Lemma 7.22. *For every integer n , $n \in \{6, 7, 9, 10\}$, D_n is $(3, 4, 3^{n-4}, 4, 3)$ -choosable. For $n \in \{8, 11\}$, D_n is $(3, 3, 3, 5, 3^{n-7}, 2, 4, 3)$ -choosable.*

Proof. To prove the lemma, we again use Theorem 1.77 and for each $n \in \{6, 7, 8, 9\}$, we provide a monomial m_n such that $d_n(m_n) \neq 0$ in D_n .

- $d_6(X_1^2 X_2^3 X_3^2 X_4^2 X_5^2 X_6^2) = 2$,
- $d_7(X_1^2 X_2^3 X_3^2 X_4^2 X_5^2 X_6^2 X_7^2) = 1$,
- $d_8(X_1^2 X_2^2 X_3^2 X_4^3 X_5^2 X_6 X_7^3 X_8^2) = -1$,
- $d_9(X_1^2 X_2^3 X_3^2 X_4^2 X_5^2 X_6^2 X_7^2 X_8^2 X_9^2) = 2$,
- $d_{10}(X_1^2 X_2^3 X_3^2 X_4^2 X_5^2 X_6^2 X_7^2 X_8^2 X_9^2 X_{10}^2) = 1$,
- $d_{11}(X_1^2 X_2^2 X_3^2 X_4^3 X_5^2 X_6^2 X_7^2 X_8^2 X_9 X_{10}^3 X_{11}^2) = -1$.

It is easy to verify that the degree of every variable is less than the number of available colors assumed by the lemma, and thus we infer the desired choosabilities of D_n . \square

7.4.2 Proof

Recall that in Theorem 7.5.4, we assume the graph is class I. In our proof, this is an important feature which enables us to confirm Conjecture 7.4.2 for this class of graphs. We again prove a stronger version of the theorem.

Theorem 7.23. *Let G be a graph of class I. Then for every proper 3-edge-coloring π with colors a , b , and c , and for every color $\alpha \in \{a, b, c\}$ there exists a $(1, 2^7)$ -packing edge-coloring σ such that the edges of color α in π are colored with 0 in σ .*

For simplicity, we will refer to a $(1, 2^7)$ -packing edge-coloring σ obtained from a proper 3-edge-coloring π , in which the edges of color α in π are colored with 0 in σ , as an α -induced coloring σ_π^α .

Proof. We prove the theorem by contradiction. Let G be a minimal counter-example to the theorem minimizing the sum $|V(G)| + |E(G)|$. Let π be a proper 3-edge-coloring (using colors a , b , and c) and let the color a be the color class for which there is no $(1, 2^7)$ -packing edge-coloring σ (using colors in $\{0, 1, \dots, 7\}$ and 0 being the 1-color) of G such that all edges colored a in π are colored 0 in σ .

We begin by establishing some structural properties of G .

Claim 7.24. *G is simple.*

Proof. Suppose there are vertices u and v in G connected by at least two parallel edges. Remove one of the edges, call it e , between them (if possible, take the one colored with a) to obtain a smaller graph G' . By minimality of G , there is an a -induced coloring σ_π^a of G' . If e is colored by a in π , then extend σ_π^a by coloring e with 0. Otherwise there is no edge colored a between u and v . Therefore, if u' and v' are the respective other neighbors of u and v , when they exist, then uu' and vv' are colored a . Thus, $A_2(e) \geq 2$ and we can extend σ_π^a to G , a contradiction. Loops can be treated similarly. \blacksquare

Claim 7.25. *G is cubic.*

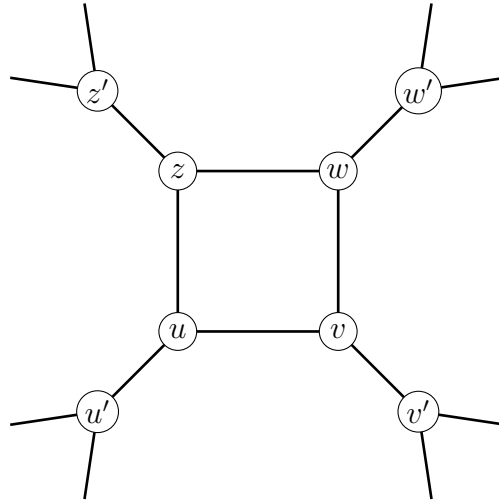


Figure 7.8: A 4-cycle with its neighborhood in G .

Proof. Suppose the contrary and let u be a 2^- -vertex. By minimality of G , there is an a -induced coloring σ_π^a of $G - u$.

Suppose first that u is a 1-vertex with a neighbor v . If uv is colored with a in π , then we color uv with 0 and hence extend σ_π^a to all the edges of G . If uv is not colored a in π , then there are at least three available 2-colors for uv and σ_π^a can be extended to G .

So, we may assume that u is a 2-vertex and let v and w be its two neighbors. We consider two subcases. First, if in π none of uv and uw is colored with a , then $A_2(uv) \geq 2$ and $A_2(uw) \geq 2$, and so we can color the two edges. Second, if in π one of uv and uw is colored with a , say uv , then we color uv with 0 and we obtain $A_2(uw) \geq 1$. Hence we can extend σ_π to all the edges of G . ■

Recall that G being cubic implies that in π every color appears at every vertex.

Claim 7.26. G does not contain triangles.

Proof. Suppose the contrary and let $C = uvw$ be a triangle in G . Let u' , v' , and w' be the neighbors of u , v , and w , respectively, not on C . Since G is cubic, by Claim 7.25, u' , v' , and w' are 3-vertices and in the coloring π exactly one of the edges uu' , vv' , and ww' is colored with a , say uu' . By minimality, there is a coloring σ_π^a of $G \setminus E(C)$ induced by π . We can extend it to the edges of C in the following way. First, we color vw with 0. Next, observe there are at least two available 2-colors for each of the edges uv and uw , hence we can always color them, a contradiction. ■

Claim 7.27. G does not contain 4-cycles.

Proof. Suppose the contrary and let $C = uvwz$ be a 4-cycle in G . Let u' , v' , w' , and z' be the neighbors of u , v , w , and z , respectively, not on C (see Figure 7.8). By Claim 7.25 and Claim 7.26, the vertices u' , v' , w' , and z' are all of degree 3 and the vertices u' and w' (resp. v' and z') are distinct. Note that it is possible that $u' = w'$ (resp. $v' = z'$) but in such a case one would have even more 2-colors available to color the cycle. Hence, we may assume that u' , v' , w' , and z' are distinct. There are three non-symmetric possibilities for π to assign colors to the edges uu' , vv' , ww' , and zz' . We consider each of them separately. Before the case analysis, observe that it is not possible to have two opposite pendent edges to C colored with a , and at least one edge of the other pair of the pendent edges not colored with a , since the edges of C could not be colored with three colors.

- *The edges uu' , vv' , ww' , and zz' are all colored with a in π .* By minimality, there is an a -induced coloring $\sigma_{\pi'}^a$ of $G' = (G \setminus V(C_4)) \cup \{u'w', v'z'\}$, where π' is a proper 3-edge-coloring of G' obtained from π by coloring $u'w'$ and $v'z'$ by a . Now, consider the partial coloring of G induced by $\sigma_{\pi'}^a$ and color the edges uu' , vv' , ww' , and zz' with 0. In this way, every edge of C has at most four 2-colors in its 2-edge-neighborhood and thus at least three available 2-colors. If the union of the available colors of all three edges contains at least four colors, we can color the edges by the Hall's Theorem.

So, we may assume that all four edges have the same three available colors, say 5, 6, and 7. This implies that on the edges incident to u' and v' there are colors 1, 2, 3, and 4 (together with 0 on the two edges incident with C). The same four colors must appear on the edges incident to v' and w' . But this means that at least two pairs of edges of the same 2-color are at distance 2 in G' , and since $u'w' \in E(G')$, we have a contradiction.

- *Two edges pendent to C and one edge of C are colored with a in π , say uu' , vv' , and wz .* Consider the graph $G' = (G \setminus \{u, v\}) \cup \{u'v'\}$ and a proper 3-edge-coloring π' of G' induced by π by coloring $u'v'$ by a . By minimality, there is an a -induced coloring $\sigma_{\pi'}^a$ of G' .

Now, consider the coloring σ of G induced by $\sigma_{\pi'}^a$, where only the edges uz , vw , and uv remain non-colored. There are at most six 2-colors in the 2-edge-neighborhood of uv , so we may color it. After that, there are at most six 2-colors in the 2-edge-neighborhoods of uz and vw , since z' and w' are each incident with one edge of color 0. However, there are four distinct 2-colors incident with the vertices u' and v' (recall that $u'v' \in E(G')$), and thus the union of available colors for uz and vw contains at least two colors, meaning that we can complete the coloring, a contradiction.

- *Two edges of C are colored with a in π , say uv and wz .* Let $G' = G \setminus V(C)$ and let π' be a proper 3-edge-coloring of G' induced by π . By the minimality, there is an a -induced coloring $\sigma_{\pi'}^a$ of G' . Now, consider the coloring σ of G induced by $\sigma_{\pi'}^a$ and color the edges uv and wz with 0. For the non-colored edges, we have the following numbers of available colors: $A_2(uu') = A_2(vv') = A_2(ww') = A_2(zz') = 3$ and $A_2(uz) = A_2(vw) = 5$. To show that σ can be extended to non-colored edges, we apply Theorem 1.77 in the following way. First, associate the variables X_1, X_2, X_3, X_4, X_5 , and X_6 to the edges uu' , uz , zz' , vv' , vw , and ww' , respectively. The chromatic polynomial of a subgraph induced by the non-colored edges and setting adjacencies whenever two edges are at distance at most 2 is

$$\begin{aligned} f(X_1, \dots, X_6) &= (X_1 - X_2)(X_1 - X_3)(X_1 - X_4)(X_1 - X_5) \\ &\quad \times (X_2 - X_3)(X_2 - X_4)(X_2 - X_5)(X_2 - X_6) \\ &\quad \times (X_3 - X_5)(X_3 - X_6)(X_4 - X_6)(X_4 - X_5)(X_5 - X_6) \end{aligned}$$

Expanding the polynomial, we infer that the coefficient of the monomial $X_1^2 X_2^4 X_3^2 X_4^2 X_5^3$ in f equals +1. Thus, by Theorem 1.77, we can extend σ to all the edges of G , a contradiction.

This establishes the claim. ■

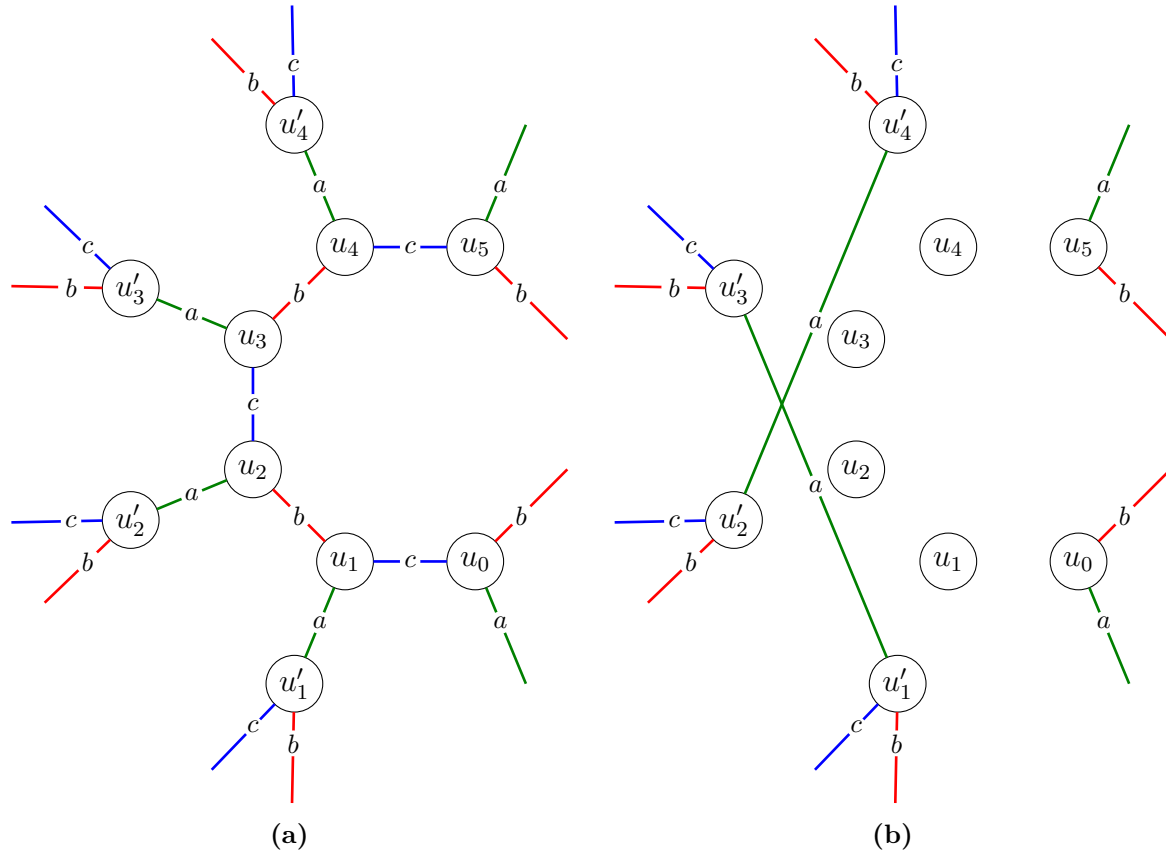


Figure 7.9: A path P on which we perform a crossing in G (a), and the configuration in G' (b).

Hence, G has girth at least 5. We continue by considering properties of longer cycles in G . First, we introduce some additional definitions. A cycle colored only with the colors b and c is called a bc -cycle. Let $P = u_0u_1u_2u_3u_4u_5$ be a path of distinct vertices on a bc -cycle. Let u'_i be the neighbor of u_i such that $u_iu'_i$ is colored with a in π for $i \in \{0, \dots, 5\}$. A P -crossing is a pair (G', π') such that $G' = G - E(P) - \{u_ju'_j\}_{1 \leq j \leq 4} + \{u'_1u'_3, u'_2u'_4\}$ and π' is obtained from π by coloring $u'_1u'_3$ and $u'_2u'_4$ with the color a (see Figure 7.9). By minimality of G , there exists an a -induced coloring $\sigma_{\pi'}^a$ of G' . The partial coloring $\sigma_{\pi'}^a$ of G induced by $\sigma_{\pi'}^a$ leaves uncolored the edges of P and the four edges $u_ju'_j$ for $1 \leq j \leq 4$. Clearly, we can color the latter four edges with 0, and so only the edges of P need to be colored. In the next claim, we give a useful property about their lists of available colors.

Claim 7.28. *Let L_i be the set of 2-colors available for the edge u_iu_{i+1} of P , where $1 \leq i \leq 3$. In the coloring $\sigma_{\pi'}^a$, one of the following properties hold:*

1. $|L_2| = 5$, or
2. there exists a color $x \in L_2$ such that $|L_1 \setminus x| \geq 3$ and $|L_3 \setminus x| \geq 3$, or
3. $|L_2| \geq 4$ and $L_1 \cap L_3 = \emptyset$. Moreover, there is a color $x \in L_1$ such that $x \notin L_2$, and a color $y \in L_3$ such that $y \notin L_2$.

Proof. Note first that the edges u_1u_2 , u_2u_3 , and u_3u_4 all have four colored edges in their 2-edge-neighborhood. Without loss of generality, we may assume that u'_1 is incident with edges colored 1 and 2, u'_3 is incident with edges colored 3 and 4 (the colors are distinct as

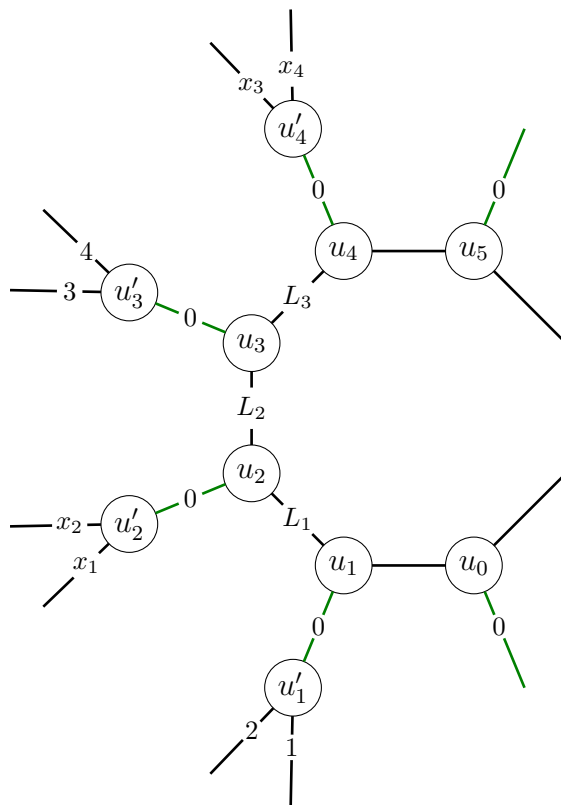


Figure 7.10: Configuration in G for considering properties of L_1 , L_2 , and L_3 .

u'_1 and u'_3 are adjacent in G'). Denote by $S_1 = \{x_1, x_2\}$ the set of 2-colors on the edges incident with u'_2 , and by $S_2 = \{x_3, x_4\}$ the set of 2-colors on the edges incident with u'_4 (see Figure 7.10). Again, $S_1 \cap S_2 = \emptyset$.

We consider possibilities regarding the sets S_1 and S_2 . Note that if $|L_1| \geq 4$ and $|L_3| \geq 4$, then we always can choose such a color in L_2 to satisfy condition (2). Suppose first that $S_1 = \{1, 2\}$. Then $|L_1| = 5$ and $L_2 = \{5, 6, 7\}$. Since S_2 does not contain 1 nor 2, $\{1, 2\} \subset L_3$. Thus, either $|L_3| \geq 4$ or L_2 contains a color which is not in L_3 .

Next, suppose that $S_1 = \{1, 3\}$. Then $|L_1| \geq 4$. If $4 \in S_2$, then also $L_3 \geq 4$ and we are done. Otherwise, by symmetry, we may assume that $x_3 = 5$ and $x_4 \in \{2, 6\}$. Observe that $x_4 \in L_2$ and $x_4 \notin L_3$, thus setting $x = x_4$ gives us condition (2) of the claim.

Suppose now that $S_1 = \{1, 5\}$. Again, $|L_1| \geq 4$. If $3 \in S_2$, then $L_3 \geq 4$ and we are done. Thus, we may assume $x_3 = 6$ and $x_4 \in \{2, 7\}$. As above, observe that $x_4 \in L_2$ and $x_4 \notin L_3$, thus setting $x = x_4$ gives us condition (2).

If $S_1 = \{3, 4\}$, we are done as $|L_2| = 5$.

If $S_1 = \{3, 5\}$, we have $L_1 = \{4, 6, 7\}$ and $L_2 = \{1, 2, 6, 7\}$. If $1 \in S_2$ or $2 \in S_2$, then we set $x = 2$ or $x = 1$, respectively, to obtain condition (2). Thus, we may assume $x_3 = 6$ and $x_4 \in \{4, 7\}$. If $x_4 = 4$, then we set $x = 1$, and if $x_4 = 7$, then $L_3 = \{1, 2, 5\}$, and we have condition (3).

Finally, if $S_1 = \{5, 6\}$, then, by symmetry, $S_2 = \{1, 2\}$, and setting $x = 1$ gives us condition (2). This completes the proof. ■

Our goal is now to show that G contains no bc -cycles.

Claim 7.29. *There is no bc -cycle with chords in G .*

Proof. Suppose the contrary and let C be a bc -cycle with a chord in G . Let $P = u_0 \dots u_n$ be a path of C such that $u_1 u_{n-1}$ is a chord of C and P is the shortest path with this property in C . For every $i \in \{0, \dots, n\}$, denote by u'_i the neighbor of u_i such that $\pi(u_i u'_i) = a$. Note that $\pi(u_i u'_i) = a$ implies that all u'_i s are pairwise distinct. Note that by definition $u'_1 = u_{n-1}$ and $u'_{n-1} = u_1$. We split the proof in two cases regarding the length of P .

Suppose first that $n \geq 12$. Let $P' = u_2 u_3 u_4 u_5 u_6 u_7$ and let (G', π') be the P' -crossing and σ_π^a an a -induced coloring of G' . Let σ_π^a be the partial coloring of G induced by σ_π^a . For every $i, i \in \{3, 4, 5, 6\}$, color $u_i u'_i$ with 0. Now, only the edges of P' are non-colored. To extend σ_π^a to all edges of G , we first uncolor the edges of P that are already colored. Next, for every $i, 0 \leq i \leq n-1$, denote by L_i the list of available 2-colors of the edge $u_i u_{i+1}$ in G . Note that $|L_0| \geq 3, |L_1| \geq 4, |L_{n-2}| \geq 4, |L_{n-1}| \geq 3$, and $|L_j| \geq 3$ for $2 \leq j \leq n-3$. By the minimality of P the 2-edge-neighborhood of an edge e of P contains the same non-colored edges as in the graph D_n obtained from P by adding the edge $u_1 u_{n-1}$. Therefore, it suffices to color D_n using the lists L_i to extend σ_π^a .

To show that we can color D_n , we make use of Claim 7.28 applied to P' . In the case (1), *i.e.*, if $|L_4| = 5$, we can color D_n by Lemma 7.19, a contradiction. In the case (2), *i.e.*, if there exists a color $x \in L_4$ such that $|L_3 \setminus x| \geq 3$ and $|L_5 \setminus x| \geq 3$, we proceed as follows. We first color $u_4 u_5$ with x . We obtain, after updating the lists, the following: $|L_2| \geq 2, |L_3| \geq 3, |L_5| \geq 3, |L_6| \geq 2$. Note that since $u_4 u_5$ is already colored, we can simply assume $|L_4| \geq 5$ to be able to apply Lemma 7.19 on the other edges of D_n . Hence, we again extend σ_π^a to all edges of G , a contradiction. In the case (3), *i.e.*, if $|L_4| \geq 4$ and $L_3 \cap L_5 = \emptyset$, we use the color $x \in L_3$, which is not in L_4 , to color $u_3 u_4$. By doing this, we only decrease the number of available colors in L_1 and L_2 to 3 and 2, respectively. Next, we consecutively color $u_2 u_3, u_1 u_2$, and $u_0 u_1$. By doing this, we obtain $|L_4| \geq 3, |L_{n-2}| \geq 2$, and $|L_{n-1}| \geq 1$. All the other non-colored edges still have at least three available colors, and hence we can extend σ_π^a by coloring consecutively the edges $u_{n-1} u_n, u_{n-2} u_{n-1}, \dots, u_4 u_5$, a contradiction.

Hence, we may assume $n < 12$. By Claim 7.27, we also have that $n \geq 7$. We first consider the case when $n \notin \{8, 11\}$. Let $P' = u_1 u_2 u_3 u_4 u_5 u_6$ and let (G', π') be the P' -crossing and σ_π^a an a -induced coloring of G' . Let σ_π^a be the partial coloring of G induced by σ_π^a . For every $i, i \in \{2, 3, 4, 5\}$, color $u_i u'_i$ with 0, and uncolor the colored edges of P . It is easy to see that all the edges have at least three available colors and the edges $u_1 u_2$ and $u_{n-2} u_{n-1}$ have at least four. Note that the edges of P together with the edge $u_1 u_{n-1}$ form the graph D_n , which is $(3, 4, 3^{n-4}, 4, 3)$ -choosable by Lemma 7.22 for $n \in \{6, 7, 9, 10\}$. Thus we can extend σ_π^a to G , a contradiction.

So, we may assume $n \in \{8, 11\}$. Let $P' = u_1 u_2 u_3 u_4 u_5 u_6$ and let (G', π') be the P' -crossing and σ_π^a an a -induced coloring of G' . Let σ_π^a be the partial coloring of G induced by σ_π^a . For every $i, i \in \{2, 3, 4, 5\}$, color $u_i u'_i$ with 0. Now, only the edges of P' are non-colored. To extend σ_π^a to all edges of G , we first uncolor the edges of P that are already colored. As above, it suffices to find a list coloring of the graph D_n obtained from P by adding the edge $u_1 u_{n-1}$. We will again apply Claim 7.28. Suppose first that $|L_3| = 5$. Then, we can extend the coloring by Lemma 7.22, saying that D_n is $(3, 3, 3, 5, 3^{n-7}, 2, 3, 4)$ -choosable. Suppose now that there exists $x \in L_3$ such that $|L_2 \setminus x| \geq 3$ and $|L_4 \setminus x| \geq 3$. We color $u_3 u_4$ with x , we obtain $|L_1| \geq 3, |L_2| \geq 3, |L_4| \geq 3, |L_5| \geq 2$, and we can assume that $|L_3| \geq 5$, since it is already colored anyway. Hence, we can color D_n by Lemma 7.22, a contradiction. Finally, suppose that $|L_2| \geq 4$ and $L_1 \cap L_3 = \emptyset$. In this case, we color

the edges of P as follows. First, color u_1u_2 with a color that is not contained in L_7 . Next, color u_0u_1 and then u_2u_3 . Note that at this point, $|L_3| \geq 2$, $|L_4| \geq 3$, $|L_5| \geq 3$, $|L_6| \geq 2$, and $|L_7| \geq 2$. This means that we can complete the coloring by using Lemma 7.17(a), a contradiction. ■

Claim 7.30. *There is no bc -cycle in G .*

Proof. Suppose, to the contrary, that $C = u_1u_2 \dots u_\ell$ is a bc -cycle of length ℓ in G . By Claim 7.29, we already have that C is chordless. Clearly, ℓ is even and by Claim 7.27, $\ell \geq 6$. For every i , $1 \leq i \leq \ell$, let u'_i be the neighbor of u_i such that $\pi(u_iu'_i) = a$ (and thus all u'_i are distinct). Since C is chordless, no u'_i is a vertex of C . We consider three cases regarding ℓ .

- $\ell \equiv 0 \pmod{3}$. Let $G' = G - E(C)$. By minimality of G , there exists an a -induced coloring σ_π^a of G' . To extend σ_π^a to all edges of G , we only need to color the edges of C . Since C is chordless, the only conflicts among its edges are those generated by C . Hence, every edge of C has at least three available colors, by Theorem 7.7, every cycle of length divisible by 3 is 3-choosable. Thus, we can extend σ_π^a to G .
- $\ell \equiv 2 \pmod{3}$ (and so $\ell \geq 8$). In this case, we perform two crossings at the same time, one with the path $P = u_\ell u_1 u_2 u_3 u_4 u_5$, and the other with the path $P' = u_4 u_5 u_6 u_7 u_8 u_j$, where $j = 1$ if $\ell = 8$, and $j = 9$ otherwise. Note that the properties for the lists of available colors guaranteed in Claim 7.28 still hold. Let

$$G' = G - E(C) - \{u_i u'_i\}_{1 \leq i \leq 8} + \{u'_1 u'_3, u'_2 u'_4, u'_5 u'_7, u'_6 u'_8\}.$$

By the minimality, there exists an a -induced coloring σ_π^a of G' ; thus $u'_1 u'_3$, $u'_2 u'_4$, $u'_5 u'_7$, and $u'_6 u'_8$ are colored with 0 in σ_π^a . Without loss of generality, we may assume that the two 2-colors incident with u'_5 are 1 and 2, and that the two 2-colors incident with u'_7 are 3 and 4. Denote by S the set containing the two 2-colors incident with u'_6 . Moreover, for every i , $1 \leq i \leq \ell - 1$, let L_i be the set of available 2-colors for the edge $u_i u_{i+1}$ (and the list for the edge $u_\ell u_1$ we denote by L_ℓ). As in the previous case, we have $|L_i| \geq 3$ for every i .

We consider two possibilities regarding S . Suppose first that $3 \notin S$. Then, $3 \in L_5$ and we color $u_5 u_6$ with 3. Since u'_7 is incident with an edge colored with 3, the sizes of L_6 and L_7 do not decrease. Now, consider the lists of available colors for the edges of P . By Claim 7.28, we have three possibilities. Suppose first that $|L_2| = 5$. In this case, we may color it last, since it will have at least one available color after all the edges at distance 2 on C are colored, thus we may ignore it for now. We color $u_4 u_5$ (decreasing the size of L_6 by one) and $u_3 u_4$ (decreasing the size of L_1 by one). It remains to color the edges of the path $u_6 u_7 \dots u_\ell u_1 u_2$, which is possible by Lemma 7.17, and finally coloring L_2 , a contradiction. Suppose next that there exists $x \in L_2$ such that $|L_1 \setminus x| \geq 3$ and $|L_3 \setminus x| \geq 3$. In this case, color $u_2 u_3$ with x . We obtain $|L_1| \geq 3$, $|L_3| \geq 2$, $|L_4| \geq 1$, $|L_6| \geq 3$, \dots , $|L_{\ell-1}| \geq 3$, and $|L_\ell| \geq 2$. Color $u_4 u_5$ and then $u_3 u_4$. It remains to color the path $u_6 \dots u_\ell u_1 u_2$ (of length $\ell - 4 \equiv 1 \pmod{3}$), which can be done by Lemma 7.17(a), a contradiction. Finally, suppose that $|L_2| \geq 4$ and $L_1 \cap L_3 = \emptyset$. In this case, we again color $u_4 u_5$ (decreasing $|L_6|$ and $|L_2|$ by one) and $u_3 u_4$ (only decreasing $|L_2|$ by one). The remaining edges are lying on a path which is strongly edge-colorable by Lemma 7.17, a contradiction.

Therefore, by symmetry, we may assume $S = \{3, 4\}$. Moreover, by the same reasoning, the set of 2-colors incident with u'_2 is the same as the set of 2-colors incident with u'_3 . This altogether means that L_2 and L_6 both have size at least five, and thus we can color the edges u_2u_3 and u_6u_7 last. We first color u_3u_4 , u_4u_5 , and u_5u_6 (decreasing $|L_7|$ and $|L_1|$ by one), and then color the path $u_7 \dots u_\ell u_1 u_2$ by Lemma 7.17. Finally, color u_2u_3 and u_6u_7 , a contradiction.

- $\ell \equiv 1 \pmod{3}$ (and so $\ell \geq 10$). As in the previous case, we perform two crossings at the same time, one with the path $P = u_\ell u_1 u_2 u_3 u_4 u_5$, and the other with the path $P' = u_5 u_6 u_7 u_8 u_9 u_j$, where $j = 1$ if $\ell = 10$, and $j = 11$ otherwise. Let

$$G' = G - E(C) - \{u_i u'_i\}_{1 \leq i \leq 4} - \{u_i u'_i\}_{6 \leq i \leq 9} + \{u'_1 u'_3, u'_2 u'_4, u'_6 u'_8, u'_7 u'_9\}.$$

By minimality of G , there exists an a -induced coloring σ_π^a of G' ; thus $u'_1 u'_3$, $u'_2 u'_4$, $u'_6 u'_8$, and $u'_7 u'_9$ are colored with 0 in σ_π^a . Without loss of generality, we may assume that the two 2-colors incident with u'_6 are 1 and 2, and the two 2-colors incident with u'_8 are 3 and 4. Denote by S the set containing the two 2-colors incident with u'_7 . Moreover, for every i , $1 \leq i \leq \ell - 1$, let L_i be the set of available 2-colors for the edge $u_i u_{i+1}$ (and the list for the edge $u_\ell u_1$ we denote by L_ℓ). Again, we have $|L_i| \geq 3$ for every i .

We consider two possibilities regarding S . Suppose first that $3 \notin S$. Then, $3 \in L_6$ and we color $u_6 u_7$ with 3. Since u'_8 is incident with an edge colored with 3, the sizes of L_7 and L_8 do not decrease. Now, consider the lists of available colors for the edges of P . By Claim 7.28, we have three possibilities. Suppose first that $|L_2| = 5$. In this case, we may color it last, since it will have at least one available color after all the edges at distance 2 on C are colored, thus we may ignore it for now. We first consecutively color the edges $u_5 u_6$, $u_4 u_5$, and $u_3 u_4$ (each of them has at least one available color when being colored), by that, we decrease the sizes of L_1 and L_7 by at most 1, and hence we can color the edges of the path $u_7 \dots u_\ell u_1 u_2$ by Lemma 7.17. Finally, we color $u_2 u_3$, a contradiction. Suppose next that there exists $x \in L_2$ such that $|L_1 \setminus x| \geq 3$ and $|L_3 \setminus x| \geq 3$. In this case, color $u_2 u_3$ with x , and then consecutively $u_4 u_5$, $u_5 u_6$, and $u_3 u_4$. It remains to color the edges of the path $P'' = u_7 \dots u_\ell u_1 u_2$, where every edge has at least three available colors, except for the edges $u_1 u_2$, $u_\ell u_1$, and $u_7 u_8$, which have at least two. Since the length of P'' is $\ell - 5 \equiv 2 \pmod{3}$, we can color its edges by Lemma 7.17(a), a contradiction. Finally, suppose that $|L_2| \geq 4$ and $L_1 \cap L_3 = \emptyset$. In this case, we first color $u_5 u_6$ (decreasing $|L_7|$ by one), $u_4 u_5$, $u_3 u_4$ (not decreasing $|L_1|$), and $u_2 u_3$ (decreasing $|L_1|$ and $|L_\ell|$ by one). The remaining edges are lying on a path which is strongly edge-colorable by Lemma 7.17(a), a contradiction.

Therefore, by symmetry, we may assume $S = \{3, 4\}$. Moreover, by the same reasoning, the set of 2-colors incident with u'_2 is the same as the set of 2-colors incident with u'_3 . This altogether means that L_2 and L_7 both have size at least 5, and thus we can color the edges $u_2 u_3$ and $u_7 u_8$ last. Now, color $u_3 u_4$, $u_4 u_5$, $u_5 u_6$, and $u_6 u_7$ in this order (decreasing $|L_1|$ and $|L_8|$ by one), and then color the path $u_7 \dots u_\ell u_1 u_2$ by Lemma 7.17. Finally, color $u_2 u_3$ and $u_7 u_8$, a contradiction. ■

Since G is cubic by Claim 7.25, the subgraph of G induced by the edges colored b or c in π is 2-regular, meaning that there must be at least one bc -cycle in G , which is in contradiction with Claim 7.30. This establishes Theorem 7.23. \square

7.5 Further Work

Conjecture 7.4 remains open, but our upper bounds are only by one 2-color off. Unfortunately, we were not able to apply the techniques we used to prove tight bounds for proper edge-coloring and strong edge-coloring of subcubic graphs, to the problems considered in this chapter. Therefore, since solving Conjecture 7.4 in the general setting seems to be challenging, we suggest in this section additional problems which arise naturally when dealing with the considered colorings. All of them are supported with computational results on graphs of small orders.

We begin with a general conjecture for strong edge-coloring.

Conjecture 7.31. *Every bridgeless subcubic graph G , not isomorphic to the Wagner graph or the complete bipartite graph $K_{3,3}$ with one edge subdivided, admits a (2^9) -packing edge-coloring.*

We proceed with an overview of results in specific graph classes and list open problems for each of them. For that, we follow the conjecture on strong edge-coloring of subcubic graphs proposed by Faudree, Gyárfás, Schelp, and Tuza [68] in 1990.

Conjecture 7.32 (Faudree, Gyárfás, Schelp and Tuza [68]). *For every subcubic graph G it holds:*

1. G admits a (2^{10}) -packing edge-coloring,
2. If G is bipartite, then it admits a (2^9) -packing edge-coloring,
3. If G is planar, then it admits a (2^9) -packing edge-coloring,
4. If G is bipartite and each edge is incident with a 2-vertex, then it admits a (2^6) -packing edge-coloring,
5. If G is bipartite of girth at least 6, then it admits a (2^7) -packing edge-coloring,
6. If G is bipartite and has girth large enough, then it admits a (2^5) -packing edge-coloring.

All the cases of the conjecture, except Case 5, are already resolved, and we present the results in what follows.

7.5.1 Planar graphs

It was the well-known connection between edge-coloring of bridgeless cubic planar graphs and the Four Color Problem, established by Tait [179], which initiated the research in this area. By the Four Color Theorem, we thus have that every bridgeless cubic planar graph admits a $(1, 1, 1)$ -packing edge-coloring. The condition of being cubic is necessary, since already K_4 with one subdivided edge is class II. However, not all questions are resolved. The following conjecture of Albertson and Haas [3], which is a special case of Seymour's conjecture [172], is still widely open.

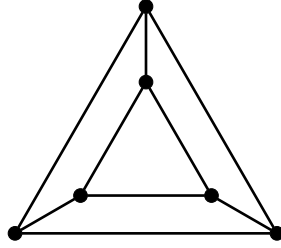


Figure 7.11: A cubic planar graph which needs nine colors for a strong edge-coloring.

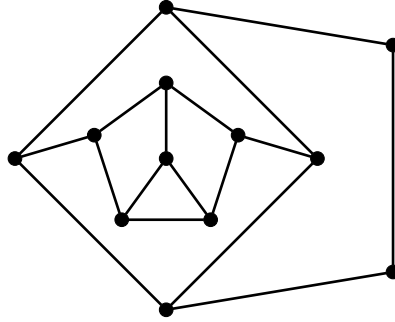


Figure 7.12: A subcubic planar graph which does not admit a $(1, 2^5)$ -packing edge-coloring nor a $(1, 1, 2^2)$ -packing edge-coloring.

Conjecture 7.33 (Albertson and Haas [3]). *Every bridgeless subcubic planar graph with at least two vertices of degree 2 admits a $(1, 1, 1)$ -packing edge-coloring.*

The number of required colors for strong edge-coloring of planar graphs is also determined. Just recently, Kostochka, Li, Ruksasakchai, Santana, Wang and Yu [133] proved the following (and resolved Conjecture 7.32.3).

Theorem 7.34 (Kostochka, Li, Ruksasakchai, Santana, Wang and Yu [133]). *Every subcubic planar graph admits a (2^9) -packing edge-coloring.*

The upper bound is tight due to the 3-prism, depicted in Figure 7.11. For now, this is the only known planar graph with maximum degree 3 with strong chromatic index equal to 9.

On the other hand, there are no results for planar graphs on the colorings with one or two matchings. We propose the following conjecture.

Conjecture 7.35. *Every subcubic planar graph admits a $(1, 2^6)$ -packing edge-coloring and a $(1, 1, 2^3)$ -packing edge-coloring.*

The conjectured upper bounds, if true, are tight and attained by infinitely many bridgeless subcubic planar graphs for both values. Indeed, in Figure 7.12, we present a planar bridgeless graph which does not admit a $(1, 2^5)$ -packing edge-coloring nor a $(1, 1, 2^2)$ -packing edge-coloring. Moreover, this graph can be appended to other subcubic graphs by the two 2-vertices, thus creating an infinite family of bridgeless subcubic planar graphs not admitting such a coloring. This conjecture also appears to be more demanding than the result of Theorem 7.34. Thus, some partial results, with additional restrictions on the structure of planar graphs, might also be interesting in order to understand the general problem better.

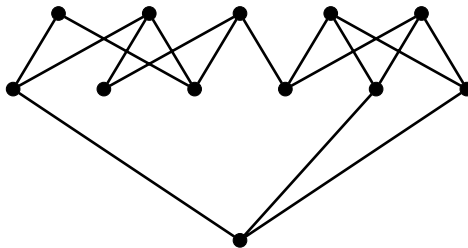


Figure 7.13: A subcubic bipartite graph which does not admit a $(1, 2^5)$ -packing edge-coloring nor a $(1, 1, 2^2)$ -packing edge-coloring.

7.5.2 Bipartite graphs

In the class of bipartite graphs, the proper and the strong case of the colorings are long solved. In 1916, König [132] proved that every bipartite graph is class I, and in 1993, Steger and Yu [177] established the following (and resolved Conjecture 7.32.2).

Theorem 7.36 (Steger and Yu [177]). *Every subcubic bipartite graph admits a (2^9) -packing edge-coloring.*

Again, the bound is tight; it is attained by, e.g., $K_{3,3}$.

Since all bipartite graphs are class I, the results and conjectures for them apply also in the bipartite case. It is known that as soon as we require some 2-colors instead just 1-colors, the problems become much harder. E.g., a tight upper bound for a strong edge-coloring of bipartite graphs is still not known (c.f. [68, 177]). Therefore, Conjecture 7.4.3 and Conjecture 7.4.4 may be considered just in the bipartite setting. Moreover, we have an infinite number of graphs attaining the conjectured upper bounds also among bipartite graphs; the bipartite graph with two 2-vertices presented in Figure 7.13 does not admit a $(1, 2^5)$ -packing edge-coloring nor a $(1, 1, 2^2)$ -packing edge-coloring, and so an infinite family of such graphs can again be constructed.

If we consider subcubic graphs with only edges of weight at most 5, i.e., edges where at least one of the end-vertices is of degree at most 2, the number of required colors decreases substantially. In particular, Conjecture 7.32.4 was resolved by Maydanskiy [152] and independently by Wu and Lin [201].

Theorem 7.37 (Maydanskiy [152], and Wu and Lin [201]). *Every subcubic bipartite graph, in which each edge has weight at most 5, admits a (2^6) -packing edge-coloring.*

Clearly, an analogous question for coloring such graphs with two 1-colors is whether they admit a $(1, 1, 2^2)$ -packing edge-coloring. It is answered in affirmative [176]. The bound is tight already in the class of trees. On the other hand, we do not have the answer for the following.

Question 7.38. *Is it true that every subcubic bipartite graph, in which each edge has weight at most 5, admits a $(1, 2^4)$ -packing edge-coloring?*

This bound is again attained in the class of trees.

7.5.3 Graphs with large girth

Similarly as the bipartiteness, having large girth does not really simplify edge-colorings in which some colors must be 2-colors. Even more, due to Kochol [131], we know that there

are graphs with arbitrarily large girth which are class II. Anyway, if the girth is infinite, *i.e.* we consider the trees, the following simple observation is immediate.

Observation 7.39. *Every subcubic tree admits:*

1. a $(1, 1, 1)$ -packing edge-coloring,
2. a $(1, 1, 2^2)$ -packing edge-coloring,
3. a $(1, 2^4)$ -packing edge-coloring,
4. a (2^5) -packing edge-coloring.

The bounds are tight already if we just consider a neighborhood of one edge with both end-vertices of degree 3.

In the case of strong edge-coloring, Conjecture 7.32.6 was also rejected just recently by Lužar, Mačajová, Škoviera, and Soták [147], who proved that a cubic graph is a cover of the Petersen graph if and only if it admits a (2^5) -packing edge-coloring.

Before we consider the intermediate colorings, we first recall the result of Gastineau and Togni [84].

Proposition 7.40 (Gastineau and Togni [84]). *Every cubic graph admitting a $(1, 1, 2^2)$ -packing edge-coloring is class I and has order divisible by four.*

Hence, the analogue of Conjecture 7.32.6 when having two 1-colors does not hold. However, the following remains open.

Question 7.41. *Is it true that every subcubic bipartite graph with large enough girth admits a $(1, 2^4)$ -packing edge-coloring?*

To conclude, we believe that studying properties of the considered edge-colorings will have impact to the initial problem of strong edge-coloring, which is in general case still widely open. Namely, the conjectured upper bound for graphs with maximum degree $\Delta(G)$ is $1.25\Delta(G)^2$, while currently the best upper bound is due to Hurley, de Joannis de Verclous, and Kang [112], set at $1.772\Delta(G)^2$.

Chapter 8

Complexity of the injective edge-coloring problem

We study the algorithmic complexity of the injective edge-coloring problem. Our aim is to determine restricted graph classes where the problem is NP-hard, while in contrast, designing polynomial algorithms for other graph classes.

Definition 8.1 (Injective edge-coloring). An *injective k -edge-coloring* of an undirected graph G is an edge-coloring of G verifying that every two edges uv and $u'v'$, for which uu' or vv' is also an edge of G , receive distinct colors. In other words, for any 3-edge path of G (possibly forming a triangle), the first and last edge of the path receive distinct colors. The *injective chromatic index* of G , denoted $\chi'_i(G)$, is the smallest integer k for which G admits an injective k -edge-coloring.

See Figure 8.1 for an example of injective 4-edge-coloring.

The concept of an injective edge-coloring is the natural edge-version of the notion of an injective vertex coloring, introduced in [89] and well-studied since then. This edge-version was recently introduced by Cardoso, Cerdeira, Cruz and Dominic in [39], where it is studied for some classes of graphs, and the question of determining the injective chromatic index of a graph was proved to be NP-complete. Bounds on the injective chromatic index of planar graphs, graphs of given maximum degree, and other important graph classes, have been recently determined in [11, 32, 73, 134, 143, 207]. In particular, as mentioned in [73], it follows from [11] that all planar graphs are injectively 30-edge-colorable, while outerplanar graphs are injectively 9-edge-colorable [73]. It is also proved in [134] that subcubic graphs are injectively 7-edge-colorable, while subcubic bipartite graphs [73] and subcubic planar graphs [134] are injectively 6-edge-colorable. Moreover all subcubic planar bipartite graphs are injectively 4-edge-colorable [134].

Note that in [11], this notion is studied as the *induced star arboricity* of a graph, that is, the smallest number of star forests into which the edges of the graph can be partitioned: this is an equivalent way to interpret injective edge-coloring (see [73]).

Injective edge-coloring is closely related to strong edge-coloring: a strong edge-coloring of a graph G is an injective edge-coloring which is also proper. It follows from the definitions that for any graph G , $\chi'_i(G) \leq \chi'_s(G)$ holds. The algorithmic complexity of determining the strong chromatic index of a graph is well-studied, see for example [106] for a classic reference, and [49, 104] for more recent ones. In this chapter, we wish to undertake similar types of studies for the injective chromatic index. The problem at hand is formally defined as follows.

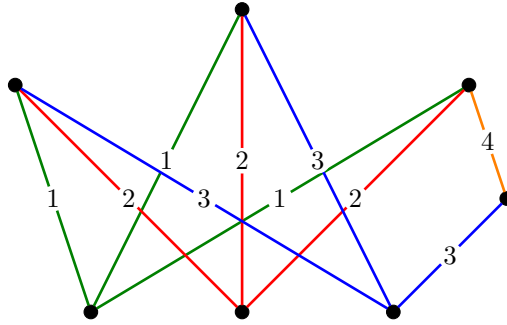


Figure 8.1: An injective 4-edge-coloring of $K_{3,3}$ with one subdivided edge.

INJECTIVE k -EDGE-COLORING

Input: A graph G .

Question: Does G admit an injective k -edge-coloring?

INJECTIVE k -EDGE-COLORING was proved NP-complete (for every fixed $k \geq 3$) in [39], with no particular restriction on the inputs. We strengthen this as follows.

Theorem 8.2. *The two following problems are NP-complete:*

1. INJECTIVE 3-EDGE-COLORING, even for triangle-free cubic graphs, and
2. INJECTIVE 4-EDGE-COLORING, even for cubic graphs.

Answering a question from [39] about the complexity of INJECTIVE k -EDGE-COLORING for planar graphs, we also study restricted subclasses of planar graphs.

Theorem 8.3. *Let $g \geq 3$. INJECTIVE 3-EDGE-COLORING is NP-complete even for:*

1. planar subcubic graphs with girth at least g ,
2. planar bipartite subcubic graphs of girth 6.

The two items in Theorem 8.3 cannot be combined, because we can prove the following (note that all planar bipartite subcubic graphs are injectively 4-edge-colorable [134]).

Theorem 8.4. *Every planar bipartite subcubic graph of girth at least 16 is injectively 3-edge-colorable.*

We also obtain an FPT algorithm when INJECTIVE k -EDGE-COLORING is parameterized by the treewidth of the input graph.

Theorem 8.5. *For every graph G of order n and every positive integer k , there exists a $2^{O(k \cdot tw(G)^2)} n$ time algorithm that solves INJECTIVE k -EDGE-COLORING.*

It is proved in [11] that $\chi'_i(G) \leq 3 \binom{tw(G)}{2}$, and so using the above algorithm, one can determine the injective chromatic index of a graph of order n in time $2^{O(tw(G)^4)} n$.

Contrasting with our hardness results for planar graphs, Theorem 8.5 implies that INJECTIVE k -EDGE-COLORING can be solved in polynomial-time on subclasses of planar graphs: K_4 -minor-free graphs (*i.e.* graphs of treewidth 2), and thus, on outerplanar graphs.

In [39], Cardoso, Cerdeira, Cruz and Dominic use a reduction on graphs having their maximum degree linear in the number of colors. We improve it with the following result.

Theorem 8.6. *For every integer $k \geq 45$, INJECTIVE k -EDGE-COLORING is NP-complete even for graphs with maximum degree at most $5\sqrt{3k}$.*

The bound of Theorem 8.6 is tight up to a constant factor: by a standard maximum degree argument of a conflict graph, every graph with maximum degree at most $\sqrt{k/2}$ is injectively k -edge-colorable. Indeed, for every edge e of a graph G , there are at most $2(\Delta(G) - 1)^2$ edges which cannot have the same color as e . Hence having at least one more color than $2(\Delta(G) - 1)^2$ allows us to color the graph.

This chapter is joint work with Florent Foucaud and Hervé Hocquard. An article version of this chapter is published in Information Processing Letters [75].

This research was supported by the IFCAM project “Applications of graph homomorphisms” (MA/IFCAM/18/39) and by the ANR project HOSIGRA (ANR-17-CE40-0022).

This chapter is divided as follows. Each of the following sections is devoted to the proof of one of our theorems. Section 8.1 contains the proof of Theorem 8.2 which deals with cubic graphs. For planar graphs, Section 8.2 presents the proofs of our results on planar graphs, namely Theorem 8.3 and Theorem 8.4. Section 8.3 contains the FPT algorithm of Theorem 8.5. Finally, Section 8.4 contains the proof of Theorem 8.6. We conclude in Section 8.5.

Contents

8.1	NP-completeness for cubic graphs	208
8.1.1	Proof of Theorem 8.2.1	208
8.1.2	Proof of Theorem 8.2.2	209
8.2	Complexity results for planar graphs	211
8.2.1	Proof of Theorem 8.3.1	211
8.2.2	Proof of Theorem 8.3.2	213
8.2.3	Proof of Theorem 8.4	214
8.3	Injective k-Edge-Coloring is FPT when parameterized by the treewidth	214
8.4	Injective k-Edge-Coloring is NP-complete even for graphs with maximum degree $O(\sqrt{k})$	216
8.5	Conclusion	218

8.1 NP-completeness for cubic graphs

For the two problems we consider, we reduce from 3-EDGE-COLORING, which is NP-complete even for cubic graphs [106].

3-EDGE-COLORING

Input: A cubic graph G .

Question: Does G admit a proper 3-edge-coloring?

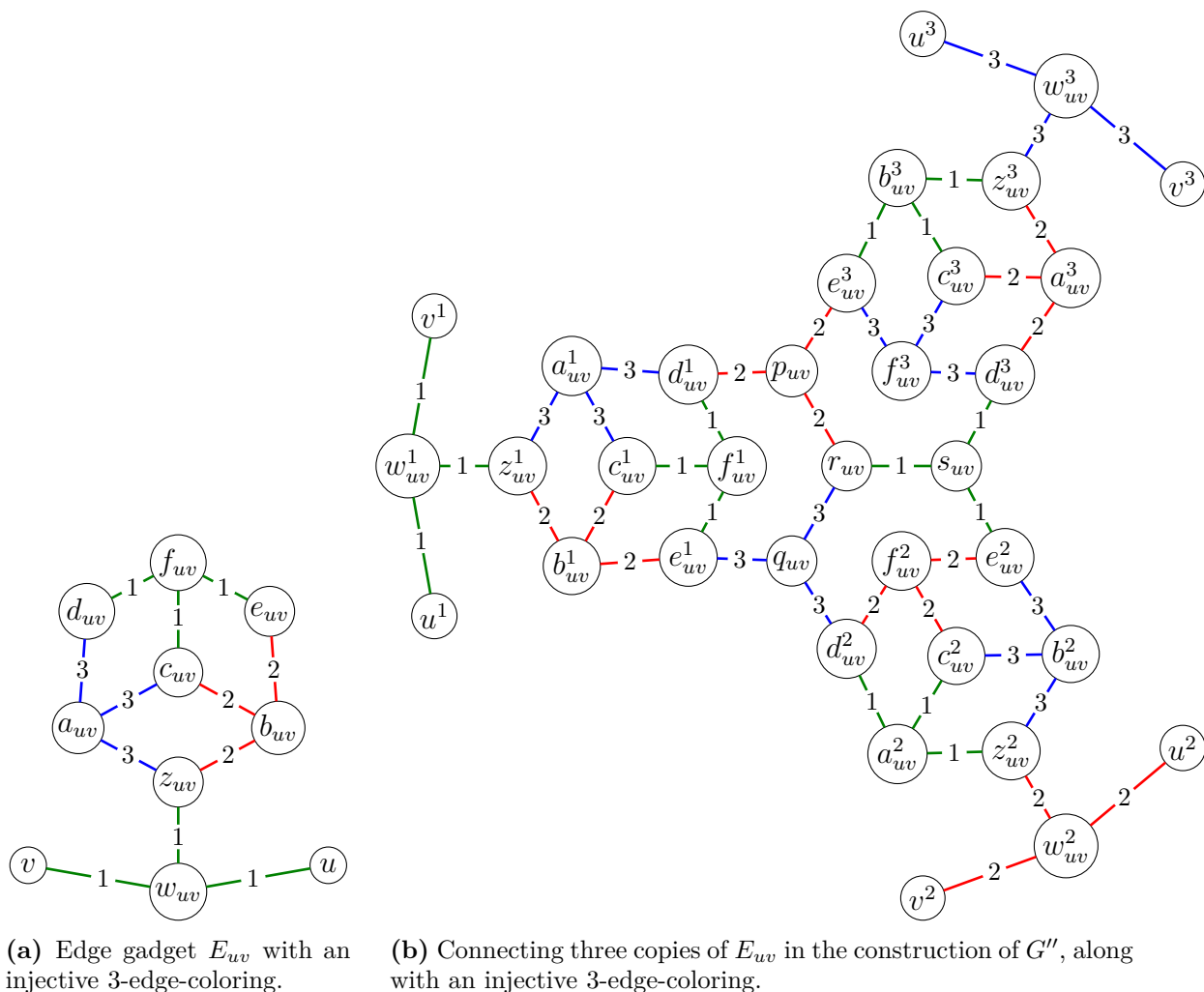


Figure 8.2: Edge gadgets used in the proof of Theorem 8.2.1.

8.1.1 Proof of Theorem 8.2.1

Proof of Theorem 8.2.1. Let G be the input cubic graph. We will proceed in two steps: first, we create a triangle-free subcubic graph G' which has an injective 3-edge-coloring if and only if G is properly 3-edge-colorable. Then we describe how to make the graph cubic.

We create the graph G' from G by removing all the edges of G . For each edge uv of G , we create a copy of a gadget E_{uv} (see Figure 8.2(a) for an illustration) and connect it to u and v as follows. We add eight new vertices $w_{uv}, z_{uv}, a_{uv}, b_{uv}, c_{uv}, d_{uv}, e_{uv}$ and f_{uv} . We create the following edges $uw_{uv}, vw_{uv}, w_{uv}z_{uv}, z_{uv}a_{uv}, z_{uv}b_{uv}, a_{uv}c_{uv}, b_{uv}c_{uv}, a_{uv}d_{uv}, b_{uv}e_{uv}, c_{uv}f_{uv}, d_{uv}f_{uv}$ and $e_{uv}f_{uv}$.

Claim 8.7. E_{uv} is injectively 3-edge-colorable, and for every injective 3-edge-coloring γ of E_{uv} , we have $\gamma(uw_{uv}) = \gamma(vw_{uv}) = \gamma(w_{uv}z_{uv})$. Moreover, for any partial injective 3-edge-coloring of E_{uv} where uw_{uv}, vw_{uv} and $w_{uv}z_{uv}$ are the only colored edges of E_{uv} and have the same color, we can extend the coloring to an injective 3-edge-coloring of E_{uv} .

Proof. Let us injectively 3-edge-color E_{uv} . W.l.o.g., we can assume that $d_{uv}f_{uv}$ is colored 1, $b_{uv}c_{uv}$ is colored 2 and $a_{uv}z_{uv}$ is colored 3. We deduce that $b_{uv}e_{uv}$ is colored 2, $c_{uv}f_{uv}$ is

colored 1, $a_{uv}d_{uv}$ and $a_{uv}c_{uv}$ are colored 3, $b_{uv}z_{uv}$ is colored 2 and $e_{uv}f_{uv}$ is colored 1. Hence uw_{uv} , vw_{uv} and $w_{uv}z_{uv}$ must all be colored 1.

Now, given one same color for these three edges, one can color the rest of the gadget, for example using the previously constructed coloring. ■

If G has a proper 3-edge-coloring γ , we injectively 3-edge-color G' by assigning to uw_{uv} , vw_{uv} and $w_{uv}z_{uv}$ in G' the color $\gamma(uv)$; then we extend the coloring to each E_{uv} using Claim 8.7.

Conversely, if G' has an injective 3-edge-coloring, then we color an edge uv of G with the color of the edge uw_{uv} (or vw_{uv}) of G' . This coloring is proper since Claim 8.7 insures that uw_{uv} and vw_{uv} have the same color. Indeed if ux is an edge adjacent to uv , then uw_{uv} and xw_{ux} have different colors.

We now show how to make the construction cubic. We create the cubic graph G'' as follows. First, take three disjoint copies G_1, G_2 and G_3 of G' . To differentiate the vertices of each copy, we add an exponent to the name of the vertex corresponding to the number of the copy. For example, vertex w_{uv} of G_1 will be noted w_{uv}^1 . For each edge uv of G , connect G_1, G_2 and G_3 via $K_{1,3}$ with vertex classes $\{r_{uv}\}$ and $\{s_{uv}, p_{uv}, q_{uv}\}$ as follows. The vertex s_{uv} (resp. p_{uv} , resp. q_{uv}) is adjacent to d_{uv}^3 (resp. d_{uv}^1 , resp. d_{uv}^2), e_{uv}^2 (resp. e_{uv}^3 , resp. e_{uv}^1) and r_{uv} (see Figure 8.2(b)). The graph G'' is simply the graph where the edge gadget is represented in Figure 8.2(b) and for each $u \in V(G)$, the three copies of u^i for $i \in \{1, 2, 3\}$ are identified.

As G is cubic, G'' is triangle-free and cubic. Note that if G'' admits an injective 3-edge-coloring, then in particular G' also admits an injective 3-edge-coloring and thus by our previous arguments, G is properly 3-edge-colorable.

If G is properly 3-edge-colorable, then we fix such a coloring $\gamma : E(G) \rightarrow \{1, 2, 3\}$. For $i \in \{1, 2, 3\}$, we color the edges incident with w_{uv}^i with the color $\gamma(uv) + i$, where the colors are considered to be taken modulo 3. Then it suffices to extend the obtained coloring to each edge gadget (see Figure 8.2). □

8.1.2 Proof of Theorem 8.2.2

Proof of Theorem 8.2.2. Let G be the input graph. For each vertex u of G , we replace it by the following vertex gadget S_u (see Figure 8.3). The gadget S_u is made of a 9-cycle $x_0^u x_1^u \dots x_8^u$ and three other vertices y_i^u ($i \in \{0, 3, 6\}$) that will be connected to the rest of the graph. We add the edges $x_1^u x_8^u$, $x_2^u x_4^u$, $x_5^u x_7^u$, $x_0^u y_0^u$, $x_3^u y_3^u$ and $x_6^u y_6^u$. For any edge-coloring γ of S_u , we note $C_i^u(\gamma) = \{\gamma(x_i^u x_{i+1}^u), \gamma(x_i^u x_{i-1}^u)\}$ where $i \in \{0, 3, 6\}$ and where the indices are taken modulo 9.

Claim 8.8. *For every injective 4-edge-coloring γ of S_u and for every $i \in \{0, 3, 6\}$, the color $\gamma(x_i^u y_i^u)$ belongs to the set $C_i^u(\gamma)$. Moreover, $C_0^u(\gamma) \cup C_3^u(\gamma) \cup C_6^u(\gamma) = \{1, 2, 3, 4\}$ and there exists a color $a \in \{1, 2, 3, 4\}$ such that for all $i \in \{0, 3, 6\}$, $a \in C_i^u(\gamma)$.*

Furthermore, for any choice of color for $x_0^u y_0^u$, $x_3^u y_3^u$, $x_6^u y_6^u$ and sets of colors $C_i^u(\gamma)$, $i \in \{0, 3, 6\}$ verifying the previous necessary conditions, there exists an injective 4-edge-coloring γ of S_u matching those choices.

Proof. Let us try to construct an injective 4-edge-coloring γ of S_u . Up to permuting the colors, we assume that $\gamma(x_0^u x_1^u) = 1$, $\gamma(x_0^u x_8^u) = 2$ and $\gamma(x_8^u x_1^u) = 3$. Note that $x_2^u x_4^u$ and $x_5^u x_7^u$ cannot both be colored 4, w.l.o.g. assume that $\gamma(x_2^u x_4^u) \neq 4$. Hence $\gamma(x_2^u x_4^u) = 2$ and $\gamma(x_2^u x_3^u) = 4$. Remark that $\gamma(x_5^u x_6^u) \neq 2$. Moreover $x_5^u x_7^u$ and $x_6^u x_7^u$ can only receive

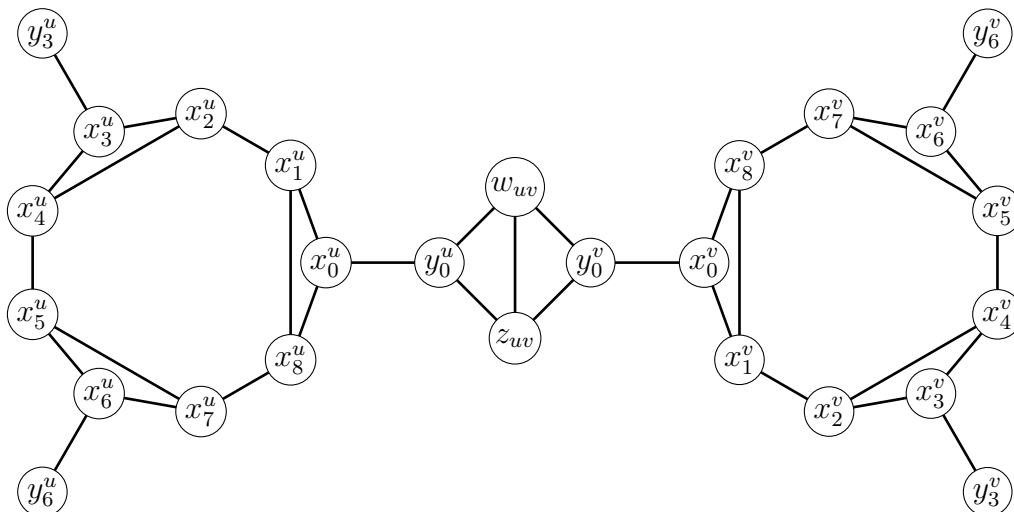


Figure 8.3: Two vertex gadgets S_u and S_v , corresponding to the vertices u and v of a graph G , connected by an edge gadget corresponding to the edge uv of G .

colors 1 or 4 and they must receive different colors. Hence $\gamma(x_5^u x_6^u) = 3$, $\gamma(x_3^u x_4^u) = 1$, $\gamma(x_5^u x_7^u) = 4$ and $\gamma(x_6^u x_7^u) = 1$. Now there are two ways to complete the coloring of S_u , either $\gamma(x_1^u x_2^u) = 4$, $\gamma(x_4^u x_5^u) = 3$ and $\gamma(x_7^u x_8^u) = 2$ or, $\gamma(x_1^u x_2^u) = 3$, $\gamma(x_4^u x_5^u) = 2$ and $\gamma(x_7^u x_8^u) = 4$. In both cases all properties of the first part of the claim hold (with $a = 1$).

Finally, note that the second of the two previous coloring options allows us to color $x_i^u y_i^u$, $i \in \{0, 3, 6\}$ with any color among those of $x_i^u x_{i+1}^u$ and $x_i^u x_{i-1}^u$, and to complete the coloring. ■

For every edge uv of G , we construct the following edge gadget E_{uv} (see Figure 8.3). First, choose y_i^u (resp. y_j^v) of degree 1 among the vertices of S_u (resp. S_v). Create two new adjacent vertices w_{uv} and z_{uv} such that $y_i^u w_{uv} y_j^v z_{uv}$ is a 4-cycle.

Claim 8.9. *For every injective 4-edge-coloring γ of G and every edge gadget E_{uv} connecting y_i^u and y_j^v ($i, j \in \{0, 3, 6\}$), we have $C_i^u(\gamma) = C_j^v(\gamma)$.*

Furthermore, any injective 4-edge-coloring γ of S_u and S_v such that $C_i^u(\gamma) = C_j^v(\gamma)$ and $\gamma(x_i^u y_i^u) = \gamma(x_j^v y_j^v)$ can be extended to an injective 4-edge-coloring of $S_u \cup E_{uv} \cup S_v$.

Proof. Suppose, w.l.o.g. by Claim 8.8, that $x_i^u x_{i+1}^u$ is colored 1, $x_i^u x_{i-1}^u$ is colored 2 and $x_i^u y_i^u$ is colored 1. Now w.l.o.g., $y_i^u w_{uv}$ is colored 3 and $y_i^u z_{uv}$ is colored 4. This implies that $w_{uv} z_{uv}$ is colored 2, $y_j^v w_{uv}$ is colored 3, $y_j^v z_{uv}$ is colored 4, $y_j^v x_j^v$ is colored 1 and $C_j^v(\gamma) = \{1, 2\}$.

The second part of the claim is proved by taking the previous coloring and extending it using the second part of Claim 8.8. ■

Let G' be the cubic graph constructed from G by the above process. By Claim 8.9, if uv is an edge connecting y_i^u and y_j^v then for any injective coloring γ of G' , $C_i^u(\gamma) = C_j^v(\gamma) = \{a, b\}$ for some a and b . Hence this set somehow characterizes the edge gadget E_{uv} , we say that E_{uv} is colored by $\{a, b\}$.

Suppose that there exists an injective 4-edge-coloring γ of G' . For each edge uv of G , we color uv depending on the coloring of E_{uv} . When E_{uv} is colored $\{1, 2\}$ or $\{3, 4\}$ (resp. $\{1, 3\}$ or $\{2, 4\}$, resp. $\{1, 4\}$ or $\{2, 3\}$) then we color uv by color 1 (resp. 2, resp. 3). We argue that this edge-coloring, noted γ , is proper. Indeed suppose it is not, then for some vertex u , w.l.o.g., uv and uw are both colored 1. This means that the coloring of G' is

such that $C_i^u(\gamma) = C_j^u(\gamma)$ or $C_i^u(\gamma) \cap C_j^u(\gamma) = \emptyset$ for some $i \neq j$ and $i, j \in \{0, 3, 6\}$. This contradicts Claim 8.8. Hence we get a proper 3-edge-coloring of G .

Conversely, suppose that there exists a proper 3-edge-coloring of G . In G' , we color each edge of the form $x_i^u y_i^u$ by 1. If an edge uv of G is colored 1 (resp. 2, resp. 3) then we assign the color $\{1, 2\}$ (resp. $\{1, 3\}$, resp. $\{1, 4\}$) to E_{uv} . By Claim 8.8, this coloring can be extended to an injective 4-edge-coloring of each S_u , $u \in V(G)$. By Claim 8.9, this injective 4-edge-coloring can be extended to each edge gadget to color the whole graph. \square

8.2 Complexity results for planar graphs

In the proof of the two results presented in Theorem 8.3, we will reduce from the following problem.

PLANAR VERTEX 3-COLORING

Input: A planar graph G with maximum degree 4.

Question: Does G admit a proper (vertex) 3-coloring?

This problem was proven to be NP-complete in [83]. Let G be a planar graph with maximum degree 4.

8.2.1 Proof of Theorem 8.3.1

Proof of Theorem 8.3.1. Recall that we want to construct a graph G' with girth at least g .

For each vertex $u \in V(G)$, we construct a vertex gadget S_u as follows (see Figure 8.4). First create a cycle $x_1^u x_2^u \dots x_\ell^u$ where $\ell \geq g$ and ℓ is an odd multiple of 3. To each x_i^u add a single pendant neighbor y_i^u of degree 1. To the vertex y_1^u , add two non-adjacent neighbors w^u and z^u . Create four more vertices a_1^u, b_1^u, c_1^u and d_1^u . The vertex w^u is adjacent to a_1^u and b_1^u while z^u is adjacent to c_1^u and d_1^u . Now construct a path $a_1^u a_2^u \dots a_g^u$ of length g and add to each a_i^u for $i \leq g - 1$ a pendant vertex of degree 1 called $a_i^{u'}$. Similarly we create the vertices $b_1^u \dots b_g^u, b_1^{u'} \dots b_{g-1}^{u'}, c_1^u \dots c_g^u, c_1^{u'} \dots c_{g-1}^{u'}$ and $d_1^u \dots d_g^u, d_1^{u'} \dots d_{g-1}^{u'}$. Finally add a vertex α^u (resp. β^u , resp. γ^u , resp. δ^u) adjacent to a_g^u (resp. b_g^u , resp. c_g^u , resp. d_g^u).

Claim 8.10. *For any injective 3-edge-coloring ρ of S_u , $\rho(a_g^u \alpha^u) = \rho(b_g^u \beta^u) = \rho(c_g^u \gamma^u) = \rho(d_g^u \delta^u)$. We call this color $\rho(S_u)$. Moreover, for any choice of a color $\rho(S_u)$, there exists an injective 3-edge-coloring ρ with these properties.*

Proof. Suppose that there exists $i \in \{1, \dots, \ell\}$ such that the property $\mathcal{P}(i) = “\rho(x_i^u x_{i+1}^u) = \rho(x_i^u y_i^u) \neq \rho(x_i^u x_{i-1}^u)”$ holds (the indices are taken modulo ℓ). Then $\mathcal{P}(i)$ holds for all $i \in \{1, \dots, \ell\}$. Indeed, take such an i , then $\rho(x_{i+1}^u x_{i+2}^u) = \rho(x_{i+1}^u y_{i+1}^u)$ is the color $\{1, 2, 3\} \setminus \{\rho(x_i^u y_i^u), \rho(x_i^u x_{i-1}^u)\}$. Hence the property holds for $i + 1$, by induction it holds for every i . Note that the same can be said for the property $\mathcal{P}'(i) = “\rho(x_i^u x_{i-1}^u) = \rho(x_i^u y_i^u) \neq \rho(x_i^u x_{i+1}^u)”$. Also note that if $\rho(x_i^u x_{i-1}^u) = \rho(x_i^u x_{i+1}^u) \neq \rho(x_i^u y_i^u)$ then we have $\mathcal{P}(i + 1)$ which is a contradiction because we do not have $\mathcal{P}(i)$.

Suppose now that for all i , neither $\mathcal{P}(i)$ nor $\mathcal{P}'(i)$ holds. This means that the edges incident to a vertex x_i^u are either of the same color, or of three distinct colors. If they have the same color, then the edges incident with x_{i+1}^u have three distinct colors, the ones incident to x_{i+2}^u have the same color, and so on. This would imply that the cycle

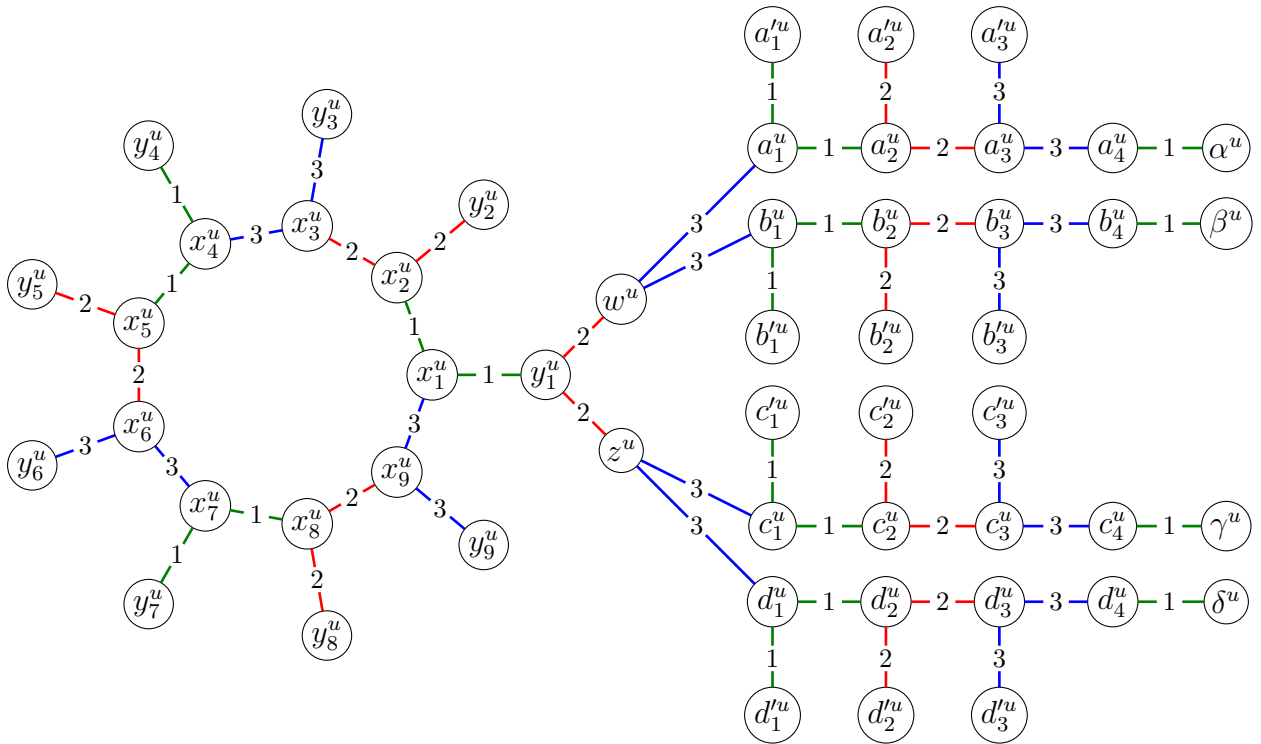


Figure 8.4: Vertex gadget S_u for planar subcubic graphs with girth at least g (in this example $g = 4$ and $\ell = 9$).

$x_1^u \dots x_\ell^u$ is even, which is a contradiction. Moreover, if the edges incident to x_i^u have three distinct colors, then the edges incident to x_{i+1}^u (or x_{i-1}^u) would all have the same color, and therefore no injective 3-edge-coloring would be possible.

Thus, w.l.o.g. we can suppose that $\rho(x_1^u x_2^u) = \rho(x_1^u y_1^u) = 1$ and $\rho(x_1^u x_\ell^u) = 3$. By extending the coloring to the rest of S_u , we can infer that $\rho(y_1^u w^u) = \rho(y_1^u z^u) = 2$, $\rho(w^u a_1^u) = \rho(w^u b_1^u) = 3$ and $\rho(z^u c_1^u) = \rho(z^u d_1^u) = 3$. By the same reasoning, we can see that all the edges of S_u (ignoring the edges involving one of the vertices x_i^u) have only one possible color which depends only on their distance to y_1^u and in particular $\rho(a_g^u \alpha^u) = \rho(b_g^u \beta^u) = \rho(c_g^u \gamma^u) = \rho(d_g^u \delta^u)$.

Conversely, S_u admits a coloring (see Figure 8.4 for an example). To choose a coloring of S_u having the desired color $\rho(S_u)$, it suffices to permute the colors in the previous coloring. ■

To finish the construction, for any edge $uv \in E(G)$, we add an edge e^{uv} to G' between a vertex among $\{\alpha^u, \beta^u, \gamma^u, \delta^u\}$ and a vertex among $\{\alpha^v, \beta^v, \gamma^v, \delta^v\}$ in such a way that the planarity of G' is preserved. This can be done by cyclically ordering the vertices of $\{\alpha^u, \beta^u, \gamma^u, \delta^u\}$ according to a planar embedding of G , and adding the edge e^{uv} between the right pair of vertices.

Note that G' is planar and subcubic with girth at least g .

Suppose that G' admits an injective 3-edge-coloring ρ . Assign to the vertex u of G the color $\rho(S_u)$. Take two adjacent vertices u and v of G . The edge e^{uv} in G' is an edge between two vertices, one of S_u and one of S_v : w.l.o.g. say $e^{uv} = \alpha^u \alpha^v$. This implies that $a_g^u \alpha^u$ and $a_g^v \alpha^v$ receive different colors and thus $\rho(S_u) \neq \rho(S_v)$. Hence this coloring of G is a proper 3-coloring.

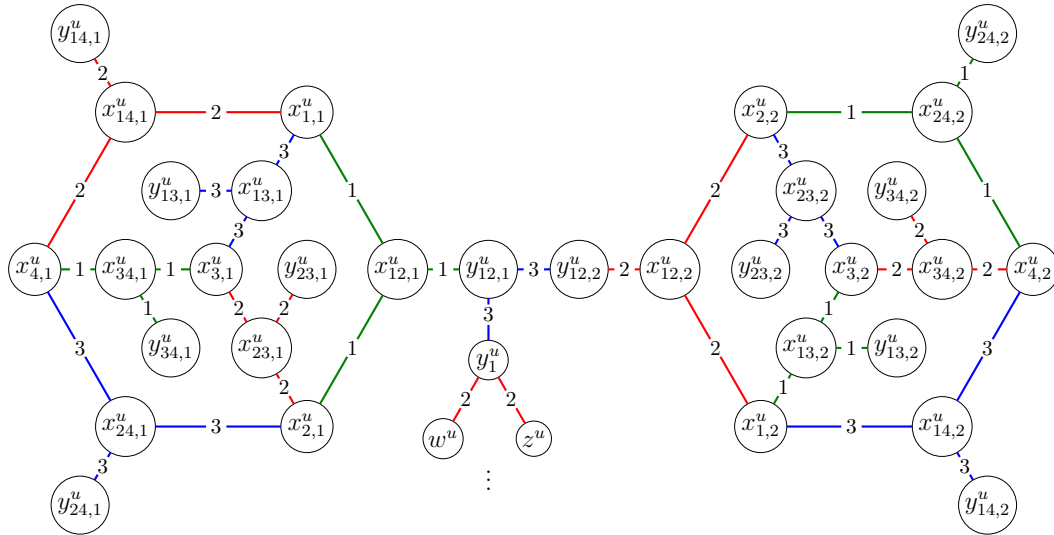


Figure 8.5: Vertex gadget for planar bipartite subcubic graphs with girth at least 6.

Conversely, suppose that G admits a proper 3-coloring. Let ρ be a partial edge-coloring of G' with no colored edges. We choose the color $\rho(S_u)$ to be the color of u in G (and we color the appropriate edges of G'). By Claim 8.10, we can extend ρ to each gadget S_u . Note that by the choice of $\rho(S_u)$, there is no conflict between edges of S_u and S_v when u and v are adjacent in G . It is left to color the edges of the form e^{uv} . By construction, there are only two edges at distance 2 of e^{uv} (and this edge does not belong to a triangle). Hence there is at least one remaining color for e^{uv} . After coloring these edges, ρ is an injective 3-edge-coloring of G' . \square

8.2.2 Proof of Theorem 8.3.2

Proof of Theorem 8.3.2. In order to prove this result, we will modify the previous construction to make it bipartite (the girth condition will be lost).

First we modify S_u (see Figure 8.5). Create the following gadget H . Start with a complete graph on four vertices x_1, \dots, x_4 . For each edge x_ix_j , create a vertex x_{ij} adjacent to both x_i and x_j and remove the edge x_ix_j . To each of these vertices of degree 2, add a pendant edge, with y_{ij} the vertex of degree 1 adjacent to x_{ij} .

We claim that in every injective 3-edge-coloring γ of H , for any $i \neq j$, the vertex x_{ij} is incident to only one color. Suppose it is not the case, then there must exist an injective 3-edge-coloring γ for which we have one of $x_{12}x_2$ and $x_{12}x_1$ colored differently from $x_{12}y_{12}$, say w.l.o.g. $\gamma(x_{12}x_1) = 1$ and $\gamma(x_{12}y_{12}) = 2$. We deduce that $\gamma(x_2x_{23}) = \gamma(x_2x_{24}) = 3$, $\gamma(x_{14}x_4) = \gamma(x_3x_{13}) = 2$, $\gamma(x_3x_{34}) = 1$, and there is no color available for $x_{23}y_{23}$, a contradiction.

Now, take two disjoint copies of H named H_1^u and H_2^u . Add an edge between the two vertices $y_{12,1}^u$ and $y_{12,2}^u$ and add the edge $y_{12,1}^u y_1^u$ where y_1^u is a new vertex. Now repeat the construction process of S_u , for $g = 6$ for example, as described in the previous section by starting at the step where the vertices w^u and z^u are added. As we observed, the edges incident to vertex $x_{12,1}^u$ of H_1^u (resp. $x_{12,2}^u$ of H_2^u) have the same color in any injective 3-edge-coloring ρ . Hence, $\rho(y_{12,1}^u y_{12,2}^u) = \rho(y_{12,1}^u y_1^u) \neq \rho(x_{12,1}^u y_{12,1}^u)$. Note that this graph also admits an injective 3-edge-coloring (see Figure 8.5). We are in the same configuration as in the proof of Theorem 2.1. Thus Claim 8.10 also holds for this gadget S_u . Note that

this gadget is bipartite.

The edge gadget does not change, it is still the edge e^{uv} . We need to be careful with the bipartiteness of the constructed graph. To ensure that the constructed graph is bipartite, it suffices that all vertices y_1^u , $u \in V(G)$, belong to the same part of the bipartition. To that end, if there is a path of odd length between y_1^u and y_1^v , then w.l.o.g. this path is $y_1^u a_1^u \dots a_g^u \alpha^u \alpha^v a_g^v \dots a_1^v y_1^v$. If we increase the length of a sequence $a_1^u \dots a_g^u$ in S_u by 3 (and also adding a_g^u , a_{g+1}^u and a_{g+2}^u), then this path now has even length. With this trick, we can ensure the bipartiteness of the constructed graph G' as well as keeping Claim 8.10 true in this new setting.

Hence, as before, G admits a proper 3-coloring if and only if G' admits an injective 3-edge-coloring. \square

8.2.3 Proof of Theorem 8.4

Proof of Theorem 8.4. Let G be a planar bipartite subcubic graph with girth at least 16. Let A and B be the two parts of the bipartition of G . We construct the graph G_A as follows: for each $u \in A$, we create a vertex u in G_A . For each pair of vertices u, v of A which are at distance 2, we add an edge between u and v in G_A . As G is subcubic, a planar embedding of G also serves as a planar embedding of G_A , where the edges of G_A follow their corresponding path of length 2 in G . Hence, G_A is a planar graph with maximum degree at most 6. Note that, by the girth condition on G , G_A does not have any k -cycle, for all k with $4 \leq k \leq 7$. Then, by the main result from [23], the graph G_A admits a 3-coloring γ .

We now color G as follows: each edge uv of G , where $u \in A$ and $v \in B$, is colored by the color $\gamma(u)$ in G_A . We claim that this is an injective 3-edge-coloring of G . Indeed, take any path $uvwz$ of G . W.l.o.g., assume $u, w \in A$ and $v, z \in B$. By construction, $uw \in E(G_A)$ and thus uv and wz receive different colors. \square

8.3 Injective k -Edge-Coloring is FPT when parameterized by the treewidth

This section presents the proof of Theorem 8.5.

Proof of Theorem 8.5. We give a fixed-parameter tractable (FPT) algorithm parameterized by the treewidth $tw(G)$ of our input graph G . We use a nice tree decomposition (see [130] and Section 1.1.9) of the input graph for our dynamic programming algorithm.

In our notation, the set of vertices of the graph associated with a node v of the tree, its bag, is denoted X_v . We call $G_{\leq v}$ the subgraph of G induced by the subtree of the decomposition rooted at v and G_v the subgraph of G induced by X_v .

We define the following set associated with a node v :

$$\mathcal{T}_v = \left\{ t_1 : X_v \rightarrow \mathcal{P}(\{1, 2, \dots, k\})^2 \right\} \times \left\{ t_2 : E(G_v) \rightarrow \{1, 2, \dots, k\} \right\},$$

where $\mathcal{P}(X)$ is the power set of X . For $T \in \mathcal{T}_v$ with $T = (t_1, t_2)$, to simplify notation, we note $T[u]$ for $t_1(u)$ when $u \in X_v$ and $T[e] = t_2(e)$ when $e \in E(G_v)$. For a vertex $u \in X_v$, we also note A_u and B_u the two sets such that $T[u] = t_1(u) = (A_u, B_u)$.

The set $\mathcal{V}_{al}(v)$ is the subset of \mathcal{T}_v such that $T \in \mathcal{V}_{al}(v)$ if and only if there exists an injective k -edge-coloring γ of $G_{\leq v}$ such that:

1. for all $u \in X_v$, $A_u = \{\gamma(uw), w \in V(G_{\leq v}) \setminus X_v\}$, i.e. A_u is the set of colors of the edges of $G_{\leq v}$ (not in G_v) incident with u ,
2. for all $u \in X_v$, $B_u = \{\gamma(zw), zw \in E(G_{\leq v}) \setminus E(G_v) \text{ and } z \in N_{G_{\leq v}}(u)\}$, i.e. B_u is the set of colors of the edges of $G_{\leq v}$ (not in G_v) at distance 2 of u (or contained in a triangle containing u),
3. for all $e \in E(G_v)$, $T[e]$ is the color $\gamma(e)$.

In this case we say that γ is *associated with* T . Note that for each injective k -edge-coloring of $G_{\leq v}$, there exists an associated $T \in \mathcal{T}_v$ and hence, $T \in \mathcal{V}_{al}(v)$. The set $\mathcal{V}_{al}(v)$ is thus the set of $T \in \mathcal{T}_v$ associated with an injective k -edge-coloring of $G_{\leq v}$.

Note that $\mathcal{V}_{al}(\text{Root}) \neq \emptyset$ if and only if there exists an injective k -edge-coloring of G . We will compute $\mathcal{V}_{al}(\text{Root})$ with a dynamic programming algorithm. Also note that $|\mathcal{T}_v| \leq 2^{O(k \cdot tw(G)^2)}$.

First suppose that v is a leaf node. Then $\mathcal{V}_{al}(v) = \mathcal{T}_v = \{(\emptyset, \emptyset)\}$.

Suppose that v is a forget node where v' is its child node such that $X_v \cup \{a\} = X_{v'}$. Let $T \in \mathcal{T}_v$, $T \in \mathcal{V}_{al}(v)$ if and only if there exists an associated coloring γ of $G_{\leq v}$. This coloring γ is also a coloring of $G_{\leq v'}$ and thus is associated with some $T' \in \mathcal{V}_{al}(v')$. In this case, since T and T' share the same coloring γ , we have the following constraints on T and T' :

- for all $e \in E(G_v)$, $T[e] = T'[e] = \gamma(e)$,
- for all $u \in X_v$ such that $au \in E(G_{v'})$, $A_u = A'_u \cup \{T[au]\}$ and $B_u = B'_u \cup \{T[aw], w \in X_v \cap N_G(a), w \neq u\}$ where $T[u] = (A_u, B_u)$ and $T'[u] = (A'_u, B'_u)$,
- for all $u \in X_v$ such that $au \notin E(G_{v'})$, we have $A_u = A'_u$ and $B_u = B'_u \cup \{T[aw], w \in X_v \cap N_G(u) \cap N_G(a)\}$ where $T[u] = (A_u, B_u)$ and $T'[u] = (A'_u, B'_u)$.

The last two constraints reflect the fact that A_u and B_u must be updated after the removal of a . The only new colors that can be added to these sets come from edges incident with a . There are multiple cases, depending on whether u and a are adjacent or not, determining which colors of edges need to be added to these sets.

Hence, for all $T \in \mathcal{V}_{al}(v)$, it suffices to check whether there exists a $T' \in \mathcal{V}_{al}(v')$ for which the previous conditions are verified. This can be done in time $2^{O(k \cdot tw(G)^2)}$, as T is uniquely determined by T' in the above constraints.

Suppose that v is an introduce node where v' is its child node such that $X_v = X_{v'} \cup \{a\}$. Let $T \in \mathcal{T}_v$, $T \in \mathcal{V}_{al}(v)$ if and only if there exists an associated coloring γ of $G_{\leq v}$. This coloring γ is also a coloring of $G_{\leq v'}$ and thus is associated with some $T' \in \mathcal{V}_{al}(v')$. In other words T is associated with a coloring γ obtained by extending a coloring γ' associated with some $T' \in \mathcal{V}_{al}(v')$. Thus $T' \in \mathcal{V}_{al}(v')$, we have the following constraints on T and T' , in order to ensure that γ is the extension of γ' :

- for all $e \in E(G_{v'})$, $T[e] = T'[e]$,
- for all $u \in X_{v'}$, $T[u] = T'[u]$,
- for $T[a] = (A_a, B_a)$, $A_a = \emptyset$ and $B_a = \bigcup_{u \in X_v, ua \in E(G_v)} A_u$,
- the coloring of X_v is an injective k -edge-coloring,
- for all $ua \in E(G_v)$, $T[ua] \notin B_u \cup \bigcup_{u' \in X_v, u' \neq u, u'a \in E(G_v)} A_{u'}$.

The first two constraints correspond to the fact that γ is an extension of γ' . As a is a new vertex, $A_a = \emptyset$ and the only colors in B_a can be obtained by edges incident with some vertex $u \in X_v$, itself adjacent to a , hence the third constraint. The last two constraints correspond to the fact that the coloring of the new edges around a cannot be in conflict with edges already colored. The fourth constraint checks that no such conflict arises in X_v and the fifth constraint ensures that for each new edge ua the color $T[ua]$ does not appear around an edge at distance 2 from a or u . For each T' , there are at most $2^{tw(G)}$ possible candidates to be added to $\mathcal{V}_{al}(v)$. Hence $2^{O(k \cdot tw(G)^2)}$ time is sufficient to compute $\mathcal{V}_{al}(v)$ from $\mathcal{V}_{al}(v')$.

Suppose that v is a join node where v_1 and v_2 are its children nodes such that $X_v = X_{v_1} = X_{v_2}$. Let $T \in \mathcal{T}_v$, $T \in \mathcal{V}_{al}(v)$ if and only if there exists an associated coloring γ of $G_{\leq v}$. As both $G_{\leq v_1}$ and $G_{\leq v_2}$ are subgraphs of $G_{\leq v}$, γ is also a coloring of $G_{\leq v_i}$ ($i \in \{1, 2\}$) and thus is associated with some $T_i \in \mathcal{V}_{al}(v_i)$. In this case, since T , T_1 and T_2 share the same coloring γ , we have the following constraints on T , T_1 and T_2 :

- for all $e \in E(G_v)$, $T[e] = T_1[e] = T_2[e]$,
- for all $u \in X_v$, $A_u = A_u^1 \cup A_u^2$ and $B_u = B_u^1 \cup B_u^2$ where $T_i[u] = (A_u^i, B_u^i)$ for $i \in \{1, 2\}$,
- for all $uw \in E(G_v)$, $A_u \cap A_w = \emptyset$.

The last constraint corresponds to the fact that the coloring is an injective k -edge-coloring (*i.e.* with no conflicts between the two subtrees). Given $T_1 \in \mathcal{V}_{al}(v_1)$ and $T_2 \in \mathcal{V}_{al}(v_2)$, T is uniquely determined by the above constraints. Hence it suffices to try all the pairs of T_1, T_2 and when the obtained set T verifies all conditions, we can add it to $\mathcal{V}_{al}(v)$. This can be done in time $(2^{O(k \cdot tw(G)^2)})^2 = 2^{O(k \cdot tw(G)^2)}$. \square

8.4 Injective k -Edge-Coloring is NP-complete even for graphs with maximum degree $O(\sqrt{k})$

We can now prove our last result: Theorem 8.6.

Proof of Theorem 8.6. We reduce from k -EDGE-COLORING, which is proven to be NP-complete even for k -regular graphs in [140].

k -EDGE-COLORING

Input: A k -regular graph G .

Question: Does G admit a proper k -edge-coloring?

We choose p to be the largest integer such that $k = \binom{p}{2} + r$ for some $r \geq 0$ (and thus $r < p$). Recall that $k \geq 45$. Moreover, we set $\ell = 2p$.

Let G be the input k -regular graph. For $uv \in E(G)$, we define the edge gadget E_{uv} as follows (see Figure 8.6). First create the following vertices $a^{uv}, b^{uv}, x_1^{uv}, \dots, x_{p-3}^{uv}, c^{uv}, d^{uv}, e^{uv}, y_1^{uv}, \dots, y_r^{uv}, s_1^{uv}, \dots, s_{2\ell-1}^{uv}$ and $s_{2\ell}^{uv}$. The vertices s_i^{uv} have degree 1 in E_{uv} and will be connected to the rest of the graph. The vertices $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}, c^{uv}\}$ form a clique; this is also the case for $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}, d^{uv}\}$ and $\{y_1^{uv}, \dots, y_r^{uv}, d^{uv}\}$. The vertex e^{uv} is adjacent to $c^{uv}, d^{uv}, x_1^{uv}, \dots, x_{p-3}^{uv}, s_1^{uv}, \dots, s_{2\ell-1}^{uv}$ and $s_{2\ell}^{uv}$. In the case where $r = 0$, *i.e.* $k = \binom{p}{2}$, we delete d^{uv} .

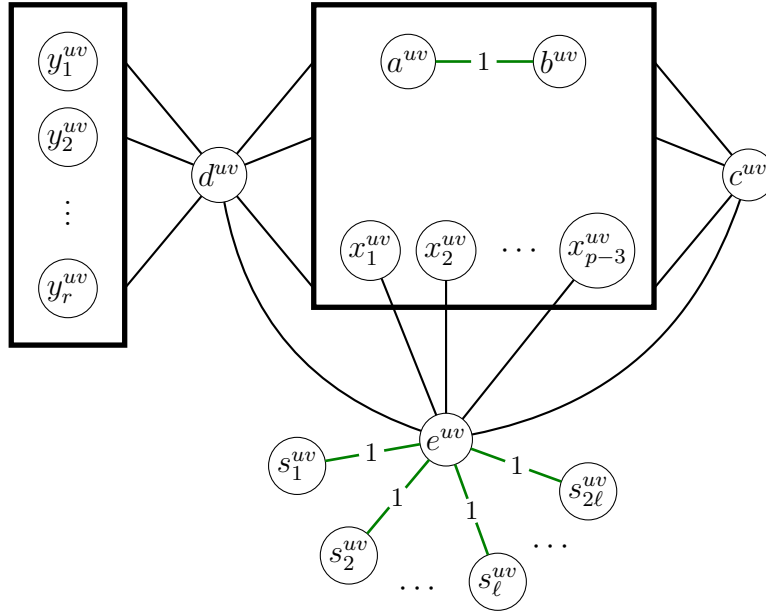


Figure 8.6: The edge gadget E_{uv} when $r > 0$. The vertices inside each of the two rectangles form a clique. The vertex c^{uv} is adjacent to every vertex inside the largest rectangle. The vertex d^{uv} is adjacent to every vertex inside the two rectangles.

Let u be a vertex of G having v_1, \dots, v_k as its neighbors. We construct the vertex gadget S_u from $k \times \ell$ vertices $v_{1,1}, \dots, v_{1,\ell}, v_{2,1}, \dots, v_{k,\ell-1}$ and $v_{k,\ell}$. We successively consider pairs v_i, v_j of neighbors. For each pair, we add an edge between one of $v_{i,1}, \dots, v_{i,\ell}$ of minimum degree and one of $v_{j,1}, \dots, v_{j,\ell}$ with minimum degree. By adding edges one by one in this way, we ensure that the maximum degree of the vertices of S_u is at most $\frac{k}{\ell} + 1$.

Finally, for each edge uv of G , we identify the 2ℓ vertices $s_1^{uv}, \dots, s_{2\ell}^{uv}$ with the ℓ vertices of S_u corresponding to v (since v is a neighbor of u , by the construction of S_u in the previous paragraph, there are ℓ such vertices in S_u) and with the ℓ vertices of S_v corresponding to u . This creates the graph G' . Note that its maximum degree is $\max(2\ell + p - 1, \frac{k}{\ell} + 2) \leq 5p \leq 5\sqrt{3k}$.

Claim 8.11. *For any injective k -edge-coloring γ of E_{uv} , we have $\gamma(e^{uv} s_1^{uv}) = \gamma(e^{uv} s_2^{uv}) = \dots = \gamma(e^{uv} s_{2\ell}^{uv})$. Moreover if γ is a partial injective k -edge-coloring of E_{uv} where $\gamma(e^{uv} s_1^{uv}) = \gamma(e^{uv} s_2^{uv}) = \dots = \gamma(e^{uv} s_{2\ell}^{uv})$ and there are no other colored edges, we can extend γ to E_{uv} .*

Proof. First note that the clique $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}, c^{uv}\}$ needs exactly $\binom{p}{2}$ distinct colors. W.l.o.g. $a^{uv}b^{uv}$ is colored 1 and the colors used for this clique are $1, 2, \dots, \binom{p}{2}$. None of these colors can be used to color the r edges of the form $d^{uv}y_i^{uv}$ hence they must be colored with $\binom{p}{2} + 1, \dots, \binom{p}{2} + r$. One can observe that an edge $e^{uv}s_i^{uv}$ cannot have a color among $\binom{p}{2} + 1, \dots, \binom{p}{2} + r$ as it is at distance 2 from the edges of the form $d^{uv}y_j^{uv}$ ($j \in \{1, \dots, r\}$). Moreover this edge cannot receive the same color as one of the edges of the clique $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}, c^{uv}\}$ except for the color 1 on the edge $a^{uv}b^{uv}$. Hence all edges of the form $e^{uv}s_i^{uv}$ have the same color.

Now suppose we have a coloring γ such that these edges $e^{uv}s_i^{uv}$ ($i \in \{1, \dots, 2\ell\}$) are all colored with the same color, say 1. We color $a^{uv}b^{uv}$ with color 1 and use the $\binom{p}{2} + r - 1$ other colors to color the rest of the edges of the clique $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}, c^{uv}\}$ and

the edges of the form $d^{uv}y_j^{uv}$ ($j \in \{1, \dots, r\}$). We color $e^{uv}z$ for $z \in \{x_1^{uv}, \dots, x_{p-3}^{uv}, c^{uv}\}$ with the color of $a^{uv}z$.

If $r = 0$, then E_{uv} is colored and γ is an injective k -edge-coloring.

If $r > 0$, we color $d^{uv}e^{uv}$ and $d^{uv}a^{uv}$ with the color of $d^{uv}y_1^{uv}$. We color $d^{uv}z$ for $z \in \{x_1^{uv}, \dots, x_{p-3}^{uv}, b^{uv}\}$ with the color of $c^{uv}z$. It is left to color the edges of the clique $\{y_1^{uv}, \dots, y_r^{uv}\}$, for which we have available the $\binom{p-1}{2}$ colors used to color the clique $\{x_1^{uv}, \dots, x_{p-3}^{uv}, a^{uv}, b^{uv}\}$, which is enough as $r \leq p-1$. This is an injective k -edge coloring of E_{uv} . ■

Suppose there is an injective k -edge-coloring γ of G' . For an edge uv of G , we color it with the color $\gamma(e^{uv}s_1^{uv})$. Take two adjacent edges of G : uv_1 and uv_2 . In S_u , there is an edge between $v_{1,i}$ and $v_{2,j}$ for some indices i and j . Thus the edges $e^{uv_1}v_{1,i}$ and $e^{uv_2}v_{2,j}$ receive different colors. By Claim 8.11, uv_1 and uv_2 receive different colors. Hence G admits a k -edge-coloring.

Suppose there is a k -edge coloring γ of G . For each edge uv , we color $e^{uv}s_i^{uv}$ with the color $\gamma(uv)$. By Claim 8.11, we can extend this coloring to all E_{uv} . At this point there is no conflict between the colored edges. Indeed the only pairs of edges which are at distance 2 and not in the same edge gadget are of the form $e^{uv}s_i^{uv}$, and since γ is proper, there is no conflict here. It is left to color the edges inside the vertex gadget. Let $e = v_{i,j}v_{i',j'}$ be an uncolored edge. As the maximum degree of the vertices of S_u is at most $\frac{k}{\ell} + 2$, there are at most $(\frac{k}{\ell} + 2)^2$ edges incident to a vertex of S_u that can be in conflict with e . We must also consider the edges incident with e^{uv_i} and e^{uv_j} . For each of the two vertices there is one forbidden color $\gamma(uv_i)$ which is common to 2ℓ edges incident to e^{uv_i} to which we need to add $p-1$ colors for the other edges of e^{uv_i} . In the end, there are at most $2p + (\frac{k}{\ell} + 2)^2$ forbidden colors for e . As $2p + (\frac{k}{\ell} + 2)^2 \leq 2p + (\frac{p-1}{4} + 2)^2 = (\frac{p-1}{4})^2 + 3p + 3 \leq k$ when $k \geq 45$ and $p \geq 10$, G' admits an injective k -edge-coloring. □

8.5 Conclusion

We proved that INJECTIVE 3-EDGE-COLORING and INJECTIVE 4-EDGE-COLORING are NP-complete on some restricted classes of subcubic graphs. One can ask whether INJECTIVE 5-EDGE-COLORING is NP-complete on subcubic graphs. A conjecture proposed by Ferdjallah, Kerdjoudj and Raspaud [73] states that every subcubic graph admits an injective 6-edge-coloring (it is proved for planar graphs in [134]). In fact, we only know of two connected subcubic graphs which require six colors: K_4 and the prism. Perhaps these are the only examples that are not 5-colorable, in which case INJECTIVE 5-EDGE-COLORING would be polynomial-time solvable for this class.

We have also proved that for planar bipartite subcubic graphs, INJECTIVE 3-EDGE-COLORING is polynomial-time solvable when the girth is at least 16 (because the answer is always YES), but NP-complete when the girth is 6. It would be interesting to determine the values of the girth of planar bipartite subcubic graphs for which INJECTIVE 3-EDGE-COLORING stays NP-complete, becomes polynomial-time solvable, and always has YES as an answer.

We also do not know whether INJECTIVE 4-EDGE-COLORING is NP-complete for bipartite subcubic graphs.

Chapter 9

Adjacent vertex-distinguishing edge coloring of graphs

In this chapter, we study adjacent vertex-distinguishing edge-colorings of graphs.

Definition 9.1 (AVD k -edge-coloring). An *adjacent vertex-distinguishing k -coloring* (AVD k -coloring for short) is a proper k -edge-coloring such that, for any two adjacent vertices u and v , the set of colors assigned to edges incident with u differs from the set of colors assigned to edges incident with v . The *AVD-chromatic index* of G , denoted by $\chi'_{avd}(G)$, is the smallest integer k such that G admits an AVD k -coloring.

It should be noted that, while an isolated edge admits no AVD coloring, the AVD-chromatic index is finite for all connected graphs on at least three vertices. AVD colorings are also known as adjacent strong edge-colorings [212], neighbor-distinguishing edge-colorings [108] or 1-strong edge-colorings [2]. Note that AVD coloring is a special case of *vertex-distinguishing proper edge-coloring*. Such a coloring is a proper edge-coloring such that no two (not necessarily adjacent) vertices are incident with the same set of colors. The corresponding chromatic index is called the *observability* and was studied for different graph classes [12, 36, 41, 69].

Since an AVD coloring is a proper edge-coloring, every graph G satisfies $\chi'_{avd}(G) \geq \Delta(G)$. In addition, every graph G with two adjacent vertices of degree $\Delta(G)$ satisfies $\chi'_{avd}(G) \geq \Delta(G) + 1$. Zhang, Liu and Wang [212] completely determined the AVD-chromatic index of paths, cycles, trees, complete graphs, and complete bipartite graphs. They noted that a cycle of length five requires five colors, but conjectured that it is the only graph with such a gap between $\chi'_{avd}(G)$ and $\Delta(G)$.

Conjecture 9.2. [212] *Every connected graph G on at least 6 vertices satisfies $\chi'_{avd}(G) \leq \Delta(G) + 2$.*

For general graphs, Hatami [95] proved that Conjecture 9.2 is off by just a constant, and Joret and Lochet improved the constant as follows.

Theorem 9.3. [120] *Every connected graph G with $\Delta(G) \geq \Delta_0$ satisfies $\chi'_{avd}(G) \leq \Delta + 19$ for some large constant Δ_0 .*

Akbari, Bidkhori and Nosrati [2] showed that $\chi'_{avd}(G) \leq 3\Delta(G)$ for every graph G without isolated edges. This bound was further improved in [211] and then in [190], the latest improvement by Vučković [187] being as follows.

Theorem 9.4. [187] *For every connected graph G without isolated edges, $\chi'_{avd}(G) \leq 2\Delta(G) + 2$.*

Balister, Györi, Lehel and Schelp [13] proved Conjecture 9.2 for connected graphs with $\Delta(G) = 3$ and for connected bipartite graphs on at least three vertices. For edge-coloring, Theorem 1.49 ensures that the chromatic index of a graph is either $\Delta(G)$ or $\Delta(G) + 1$. The classification of graphs depending on this received considerable interest (see for instance [170]). For AVD coloring, Conjecture 9.2 would imply that the AVD chromatic index of a graph can only have three values: $\Delta(G)$, $\Delta(G) + 1$ or $\Delta(G) + 2$. When considering a given graph class that allows two vertices of maximum degree to be adjacent, there are only two possible upper bounds: $\Delta(G) + 1$ or $\Delta(G) + 2$. Similarly, the classification of graph classes depending on this received subsequent interest, see for instance [47, 189] or the following results on planar graphs.

Theorem 9.5. [46, 62] *Every connected bipartite planar graph G with $\Delta(G) \geq 9$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 1$.*

Theorem 9.6. [109] *Every connected triangle-free planar graph G with $\Delta(G) \geq 12$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 1$.*

Theorem 9.7. [110] *Every triangle-free planar graph G without isolated edges satisfies $\chi'_{avd}(G) \leq \max(10, \Delta(G) + 1)$.*

However, Conjecture 9.2 remains essentially open for planar graphs. The only results obtained so far on planar graphs, are as follows.

Theorem 9.8. [108] *Every connected planar graph G with $\Delta(G) \geq 12$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 2$.*

Theorem 9.9. [204] *Every connected planar graph G with girth at least 5 which is not C_5 satisfies $\chi'_{avd}(G) \leq \Delta(G) + 2$.*

Wang and Wang [188] made the link between maximum average degree and AVD chromatic index and proved Conjecture 9.2 for graphs with $\Delta(G) \geq 3$ and $\text{mad}(G) < 3$.

Theorem 9.10. [188] *Every connected graph G with $\Delta(G) \geq 3$ and $\text{mad}(G) < 3$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 2$.*

They also gave sufficient conditions for graphs of bounded maximum average degree to be AVD $(\Delta(G) + 1)$ -colorable. Combined with results of Hocquard and Montassier [103], we have the following result.

Theorem 9.11. [103, 188] *Every connected graph G with $\Delta(G) \geq 3$ and $\text{mad}(G) < 3 - \frac{2}{\Delta(G)}$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 1$.*

A result from Hocquard and Przybyło [105], obtained for the neighbor sum distinguishing index, implies the following theorem.

Theorem 9.12. [105] *Every connected graph G with $\Delta(G) \geq 6$ and $\text{mad}(G) < 3$ satisfies $\chi'_{avd}(G) \leq \Delta(G) + 1$.*

Two main questions arise from these partial results: can this threshold of 3 as an upper-bound on $\text{mad}(G)$ be reached with a sufficiently large lower-bound on $\Delta(G)$ in the case of Theorem 9.11, and broken in the case of Theorem 9.10 and Theorem 9.12? We answer positively to these questions with Theorem 9.13: there is no threshold in the case of Theorem 9.11 (and thus in the case of Theorem 9.10 and Theorem 9.12).

Theorem 9.13. *Every connected graph G with $\Delta(G) > 3(\text{mad}(G)+1)^2$ satisfies $\chi'_{\text{avd}}(G) \leq \Delta(G) + 1$.*

In the case of edge-coloring, the best lower bound is due to Woodall [198]: every graph G with $\Delta(G) > \frac{3 \times \text{mad}(G)}{2}$ satisfies $\chi'(G) = \Delta(G)$. There is a very large gap between this bound and its AVD counterpart, but this is essentially due to the fact that most methods on edge-coloring are not transposable to AVD coloring. On the other hand, the gap between the bound for AVD coloring and its list edge counterpart is a mere constant factor [24] (note that list edge-coloring is similarly conjectured to be always possible with $\Delta(G) + 1$ colors [185]).

Since planar graphs have a maximum average degree bounded by 6, Theorem 9.13 implies that planar graphs with sufficiently large maximum degree are AVD $(\Delta(G) + 1)$ -colorable. We provide a more refined lower-bound on the maximum degree, and prove here that the triangle-free hypothesis in Theorem 9.6 is unnecessary, as is the extra color in Theorem 9.8.

Theorem 9.14. *Every connected planar graph G with $\Delta(G) \geq 12$ satisfies $\chi'_{\text{avd}}(G) \leq \Delta(G) + 1$.*

This work is joint work with Marthe Bonamy, Nicolas Bousquet and Hervé Hocquard. My coauthors first presented these results as an extended abstract at Eurocomb 2013 [21]. My contribution started in late 2019 when I was tasked to complete the proofs and the corresponding paper.

We start by presenting some basic definitions in Section 9.1. Section 9.2 presents the outline of the proofs of our two results along with the key lemmas that make them possible. Section 9.3 contains the proofs of most of these lemmas while Section 9.4 is devoted to the proof of Lemma 9.20. Finally, Section 9.5 contains the two discharging procedures to prove Theorem 9.13 and Theorem 9.14.

Contents

9.1 Preliminaries	224
9.2 Sketch of the proofs	225
9.2.1 Rephrasing Theorem 9.13 and Theorem 9.14	225
9.2.2 Reducible configurations	226
9.2.3 Discharging method	228
9.3 Proof of the reducibility lemmas	229
9.3.1 Proofs of Lemma 9.21 and Lemma 9.22	229
9.3.2 Back to other lemmas	231
9.3.3 Proof of Lemma 9.25	233
9.4 Proof of Lemma 9.20	243
9.4.1 Definitions	243

9.4.2	Overview of the proof	243
9.4.3	Recoloring algorithm	244
9.4.4	Proof of Lemma 9.31	246
9.5	Discharging procedures	252
9.5.1	The ghost vertices method for mad discharging	253
9.5.2	Proof of Theorem 9.13	253
9.5.3	Proof of Theorem 9.14	255
9.5.3.1	Discharging rules	255
9.5.3.2	Introducing a tool to simplify the case analysis	258
9.5.3.3	Back to the analysis of the discharging	261
9.6	Perspectives	264

9.1 Preliminaries

Let G be a graph. Let γ be an edge-coloring of G . We denote by $\mathcal{C}_u(\gamma)$ the set of colors in γ which appear on some edge incident with u . If there is no ambiguity, we may simply write \mathcal{C}_u . The set of colors incident with an edge uv , denoted by \mathcal{C}_{uv} , is $(\mathcal{C}_u \cup \mathcal{C}_v) \setminus \{\gamma(uv)\}$. A color c is *incident* with an edge uv if $c \in \mathcal{C}_{uv}$.

A *(partial) proper edge-coloring* is a (partial) edge-coloring such that two colored edges which share an endpoint have different colors. Two vertices u, v are in *conflict* if uv is an edge, $\mathcal{C}_u = \mathcal{C}_v$ and all the edges adjacent to exactly one of u or v are colored (intuitively, two neighbors are in conflict if no extension of the partial coloring can possibly distinguish them). Note that the edge uv does not need to be colored for u and v to be in conflict. A *(partial) safe edge-coloring* is a (partial) proper edge-coloring without conflict. Note that a total safe edge-coloring is an AVD coloring. Given a partial edge-coloring where uv is not colored, the color c *creates a conflict for uv* if u or v are in conflict with one of their neighbors when uv is colored with c . A color is *compatible with uv* if it does not create any conflict for u nor for v . Let us first state two easy observations. The first one is an immediate consequence of the definition of conflict.

Observation 9.15. *Let γ be a partial safe edge-coloring where uv is not colored. Assume that a vertex x is in conflict with u when uv is colored with c , then $d(x) = d(u)$ in G and $\mathcal{C}_x = \mathcal{C}_u \cup \{c\}$.*

In a partial safe edge-coloring where uv is the only uncolored edge, the next observation implies that each neighbor of u can be in conflict with u for at most one color of uv .

Observation 9.16. *Let γ be a partial safe edge-coloring where uv is not colored. Let c, c' be two colors not compatible for uv . Assume that a vertex x is in conflict with u when uv is colored with c and that a vertex x' is in conflict with u when uv is colored with c' . If $c \neq c'$, then $x \neq x'$.*

Proof. By Observation 9.15, we have $c \in \mathcal{C}_x = \mathcal{C}_u \cup \{c\}$ and $c \notin \mathcal{C}_{x'} = \mathcal{C}_u \cup \{c'\}$. Thus $x \neq x'$. \square

Let us finally define the *split of a vertex* in a graph. Let G be a graph and u be a vertex of degree at least 2. We define $G \otimes \{u\}$ as the graph G where u has been replaced with $d(u)$ vertices $v_1, v_2, \dots, v_{d(u)}$ of degree 1 where every v_i is adjacent to a distinct neighbor of u . By abuse of notation, we still refer to the edges in $G \otimes \{u\}$ by their name in G , *i.e.* if w is a neighbor of u in G and a neighbor of v_1 in $G \otimes \{u\}$, then we can refer to the edge wv_1 as “ wu ”.

We define similarly $G \otimes S$, where $S \subseteq V$, by deleting the edges in $G[S]$ and splitting all the elements of S successively.

For a vertex $u \in V(G)$ and an integer p , we denote by $d_p(u)$ the number of neighbors of u that are of degree p .

9.2 Sketch of the proofs

Let k be an integer. We will assume from now on that none of the graphs we consider have a vertex of degree higher than k (in other words, their maximum degree is always at most k). Due to the hypotheses of Theorem 9.13 and Theorem 9.14, we can assume that $k \geq 4$ in the rest of this chapter.

9.2.1 Rephrasing Theorem 9.13 and Theorem 9.14

For the sake of simplicity, we consider a very slight variation of AVD coloring: a proper edge-coloring where no two adjacent vertices are in conflict, unless they are both of degree 1. The two definitions are equivalent for every connected graph except K_2 : we merely state that since there is no possible AVD coloring of K_2 , we settle for a proper edge-coloring (which is always possible as soon as there at least one color in play). This allows us to remove the condition that the graph has to be on at least 3 vertices, and thus to apply induction more conveniently. From now on, AVD coloring means implicitly AVD coloring except for the connected components that are isomorphic to K_2 .

The proof of these two statements are discharging proofs. For each case, we will consider a “minimal” counter-example, and prove strong structural properties on it, too strong for such a graph to exist. The definition of “minimal” depends on an underlying partial order on graphs, which we will define differently on the class of planar graphs and on the class of graphs with bounded maximum average degree.

Let G be a graph with maximum degree at most k . In the following, $n_i(G)$ denotes the number of vertices of degree i in G (similarly for $n_{i+}(G)$ with “at least i ”). The P -sequence of G is the quadruplet $(n_{2+}(G), |E(G)|, -n_1(G), |V(G)|)$. The M -sequence of G is the sequence $(n_k(G), \dots, n_2(G), |\{v | d(v) = 2 \text{ and } v \text{ is not adjacent to a 2-vertex}\}|, n_1(G))$. The M -sequence of G is the sequence of the number of vertices of given degree, sorted by decreasing degree with as slight modification for vertices of degree 2.

Let G and G' be two graphs. The graph G is P -smaller than G' , denoted by $G \prec_P G'$ if the P -sequence of G is lexicographically smaller than the P -sequence of G' . We define M -smaller similarly to P -smaller. Note that in both cases, removing vertices, edges or splitting vertices of degree at least 2 produce a smaller graph.

A planar graph G is P -reducible to another planar graph G' if $G' \prec_P G$ and G is AVD $(k + 1)$ -colorable if G' is. A graph with maximum average degree smaller than a is M_a -reducible to another graph G' with maximum average degree smaller than a if

$G' \prec_M G$ and G is AVD $(k+1)$ -colorable if G' is. A graph G is *reducible* to G' if G is both P -reducible and M_a -reducible to G' for every $a > \text{mad}(G)$.

Note that these notions of reducibility are transitive. Indeed if G is P -reducible to H and H is P -reducible to H' , then G is P -reducible to H' , and similarly for M -reducibility. Note that a graph is reducible to the empty graph G_\emptyset iff it is AVD $(k+1)$ -colorable.

The notion of M_a -reducibility (resp. P -reducibility) induces a partial order over the set of graphs with maximum average degree at most a (resp. planar graphs). Lemmas 9.17 and 9.18 rephrase Theorems 9.13 and 9.14 with these notions of reducibility.

Lemma 9.17. *Every graph with $\text{mad}(G) < m$ and $\Delta(G) \leq k$ where $k \geq 3(m+1)^2$ is M_m -reducible to G_\emptyset .*

Lemma 9.18. *Assume $k \geq 12$. Every planar graph G with $\Delta(G) \leq k$ is P -reducible to G_\emptyset .*

In the remaining of this chapter, we prove Lemmas 9.17 and 9.18. The method is the following. First, we prove that non-reducible graphs cannot contain some structures, called reducible configurations. The reducible configurations are presented in Section 9.2.2. Then, we prove that a non-reducible graph has to be G_\emptyset . This last part consists in a discharging proof.

9.2.2 Reducible configurations

The aim of this part is to prove that a non-reducible graph cannot contain some configurations. This section is devoted to stating the reducible configurations and giving a short idea of their proofs. Complete proofs can be found in Sections 9.3 and 9.4. In this section, G denotes a simple graph such that $\Delta(G) \leq k$ and a is a real which is greater than $\text{mad}(G)$. Note that K_2 is considered to be a reducible graph.

First note that every graph which is not connected is reducible.

Lemma 9.19. *If G has a vertex v such that $G - v$ has at least two connected components which are not an isolated vertex, then G is reducible.*

The proof of Lemma 9.19 is quite simple since it just consists in AVD coloring every graph G_i obtained from a connected component C_i on at least two vertices by adding a vertex v' and vertices of degree 1 in order to emulate v , then gluing these colorings appropriately.

Lemma 9.20. *If G contains any of the following configurations, then G is reducible:*

1. Two adjacent vertices u and v with $d(v) < d(u) \leq \lfloor \frac{k}{2} \rfloor$.
2. A vertex v_2 adjacent to two vertices v_1 and v_3 , with $d(v_1) = d(v_2) = d(v_3) \leq \lfloor \frac{k-1}{2} \rfloor$.

Both statements follow from an involved recoloring algorithm. We consider an AVD coloring of the graph without the edge v_1v_2 . First we prove that we can locally modify the coloring in such a way that the set of colors adjacent to v_1 is almost the set of colors adjacent to v_2 . We then prove that, under this condition, there is an AVD coloring of the whole graph.

Before stating Lemmas 9.21 and 9.22, let us introduce some definitions and notation. Let u be a vertex and v_1, \dots, v_p be a subset of $N(u)$. The *conflict index* of v_i , denoted by

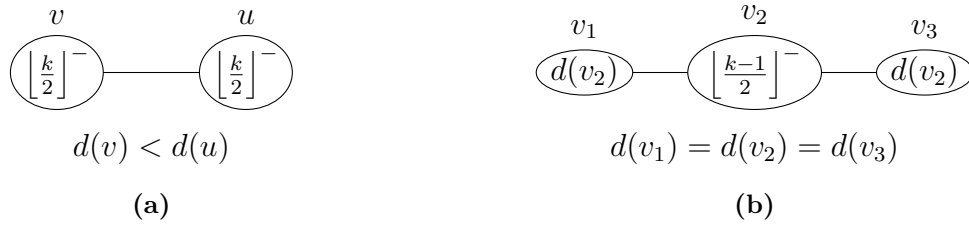


Figure 9.1: The configurations for Lemma 9.20.

$c(v_i)$, is the number of neighbors of v_i which have degree exactly $d(v_i)$. The conflict index is the maximum number of compatible colors which can create a conflict with v_i when we color uv_i .

The *recoloring index* of v_1, \dots, v_p is $\max_{i \in \{1, \dots, p\}} (d(v_i) + c(v_i) - i)$. The recoloring index roughly denotes, given a set of colors compatible with the edges incident with u , if $\{uv_1, \dots, uv_p\}$ are not colored, the maximum number of these colors which are forbidden for uv_i when the edges uv_1 to uv_{i-1} are still not colored.

Lemma 9.21. *Let v_1, \dots, v_p be p neighbors of a vertex u ($p \geq 1$) with degrees different from $d(u)$. Let $m = \max_i (d(v_i) + c(v_i) - i)$.*

If $\binom{k+1-(d(u)-p)-m}{p} \geq d_{d(u)}(u) + 1$, then every AVD $(k+1)$ -coloring α of $G - \{uv_i\}_{1 \leq i \leq p}$, where we allow u to be in conflict with other vertices, can be extended to an AVD $(k+1)$ -coloring of G . In particular, G is reducible.

Note that Lemma 9.21 is more interesting when the v_i 's are sorted in such a way that $d(v_1) + c(v_1) \leq d(v_2) + c(v_2) \leq \dots \leq d(v_p) + c(v_p)$. We will mainly use Lemma 9.21 to prove that G is reducible.

The proof of Lemma 9.21 is quite involved. However, the idea is simple: consider a coloring of $G - \{uv_i\}_{1 \leq i \leq p}$. If there are sufficiently many different sets of colors which can be put on the edges $\{uv_i\}_{1 \leq i \leq p}$ in such a way that there is no conflict between v_i and its neighbors, then one of them will not create a conflict for u . The point of this proof is to show that sufficiently many such sets exist. Let us now explain a little bit more each part of the formula. The term $(k + 1 - (d(u) - p))$ denotes the number of possible colors for edges incident with u when the p edges $\{uv_i\}_{1 \leq i \leq p}$ are not colored. The term $d(v_i) + c(v_i)$ represents the number of constraints added on the edge uv_i because of v_i . The $-i$ translates the fact that we color the edges in the decreasing order from uv_p down to uv_1 (thus uv_p has extra room compared to uv_1).

Using the same kind of arguments, we can prove the following easier statement.

Lemma 9.22. *Let v_1, \dots, v_p be p neighbors of u , of degrees different from $d(u)$, with $d(v_p) \geq 2$. If $\max_i (d(v_i) + c(v_i) - i) = 0$, then G is reducible.*

The next lemma is a consequence of Lemma 9.21 and Lemma 9.22.

Lemma 9.23. *Suppose G has a vertex u of degree at least 2 adjacent to a vertex of degree 1. If u verifies one of the following, then G is reducible:*

1. u is adjacent to a vertex of degree 2, or
2. u has at most $\lfloor \frac{k}{2} \rfloor$ neighbors of degree ≥ 3 .

Let i be an integer such that $2 \leq i < \lfloor \frac{k}{2} \rfloor$, the set D_i of a graph G is the set of vertices with degree at least 2 and at most i while B is the set of *big* vertices, *i.e.* having a degree at least $\lfloor \frac{k}{2} \rfloor + 1$.

Lemma 9.24. *If there exists an integer i such that $2 \leq i < \lfloor \frac{k}{2} \rfloor$ and $|D_i| > 4(i-1)|B|$, then G is reducible.*

The proof of this lemma is heavily inspired from a beautiful proof by Woodall [199] for a similar statement in the case of list edge-coloring with Δ colors.

In a planar embedding, a neighbor v of u is *weak* if two neighbors w_1 and w_2 of v verifies that uvw_1 and uvw_2 are triangular faces. Let us now prove that a set of configurations is reducible when we consider planar graphs.

Lemma 9.25. *For $k \geq 12$, if G is a planar graph containing any of the following configurations, then G is P -reducible.*

1. *Two adjacent 2-vertices.*
2. *A vertex adjacent to two 2-vertices.*
3. *Two adjacent vertices v_1, v_2 of degree d , where $d \in \{3, 4, 5\}$, such that the edge v_1v_2 belongs to at most one triangle.*
4. *Two adjacent 3-vertices not having two adjacent common neighbors.*
5. *A vertex adjacent to two adjacent 3-vertices and a 3^- -vertex.*
6. *A vertex adjacent to two adjacent 3-vertices and a weak 4-vertex.*
7. *A 9^- -vertex u (with $d(u) \geq 4$) adjacent to two adjacent 3-vertices.*
8. *A vertex adjacent to three 3-vertices such that two of them are weak.*
9. *A vertex adjacent to two adjacent weak 4-vertices and a 2^- -vertex.*
10. *A vertex adjacent to two pairs of adjacent 4-vertices.*
11. *A 9-vertex u with $d_9(u) \leq 4$ adjacent to two adjacent 4-vertices.*

Note that, in Lemma 9.25, the hypothesis $k \geq 12$ is not necessary for every statement. The exact lower-bound on k that is used for each statement can be found in the proofs.

The proofs of the reduction lemmas are not independent and can be found in Sections 9.3 and 9.4. Their dependence is given by the order of their statement. Nevertheless, due to the complexity of the proof of Lemma 9.20, this lemma is proved after all the other ones.

9.2.3 Discharging method

We prove both theorems by contradiction. Assume they are not true, and take a minimal counter-example (with respect to the appropriate order on graphs). As stated in Section 9.2.2, there is a list of configurations that a minimal counter-example cannot contain. We prove that a graph that does not contain any of those configurations cannot satisfy the hypothesis on the maximum average degree (or the planarity hypothesis).

To that purpose, in the case of graphs with maximum average degree less than m we assign to each vertex v a weight of $d(v) - m$, introduce discharging rules as to how

the weight can be redistributed along the graph (with conservation of the total weight of the graph), and prove that after application of the discharging rules, knowing which configurations are forbidden, every vertex has a non-negative final weight. Since no weight was created nor deleted, this implies that the average degree of the graph is at least m , hence the maximum average degree is at least m . Thus a minimal counter-example cannot satisfy the hypothesis on the maximum average degree. This completes the proof.

Similarly for planar graphs: we consider a planar embedding of the graph, and we assign to each vertex its degree minus six as a weight, and to each face two times its degree minus six. We apply discharging rules to redistribute weights along the graph with conservation of the total weight. As some configurations are forbidden, we can prove that after application of the discharging rules, every vertex and every face has a non-negative final weight. This implies that $\sum_v(d(v) - 6) + \sum_f(2d(f) - 6) = 2 \times |E(G)| - 6 \times |V(G)| + 4 \times |E(G)| - 6 \times |F(G)| \geq 0$, a contradiction with Euler's Formula which states that $|E| - |V| - |F| = -2$. Hence a minimal counter-example cannot exist.

This method of proof is called a discharging method, and was introduced in the beginning of the 20th century.

9.3 Proof of the reducibility lemmas

This section is devoted to the proof of the reducibility lemmas.

Proof of Lemma 9.19. Assume there exists a vertex v such that $G - v$ has at least two connected components which are not an isolated vertex. Let C_1 be one of them, and let C_2 be the rest of the graph. Consider the graph G_1 induced by C_1 in G where we add all the edges of the form $\{uv|u \in N(v)\}$. We define similarly the graph G_2 . Those two graphs both have maximum degree at most k . Since C_1 and C_2 both contain at least one edge, $G_1 \prec G$ and $G_2 \prec G$, so we can obtain two AVD $(k + 1)$ -colorings α_1 and α_2 of G_1 and G_2 , respectively.

Up to permuting the colors of α_2 , we can assume that every edge incident with v is colored the same in α_1 and α_2 . We then obtain a coloring β of G by coloring the edges in G_1 with α_1 , the edges in G_2 with α_2 . The only overlapping edges are the edges incident with v , which are colored the same in α_1 and α_2 . Hence, the coloring β is well-defined.

By construction, there is no conflict between two adjacent vertices in G_1 , nor in G_2 . Since there is no edge in G between a vertex in C_1 and a vertex in C_2 , it follows that β is actually an AVD $(k + 1)$ -coloring. \square

Note that Lemma 9.19 allows us to most often ignore the cases where we create a K_2 connected component when removing part of the graph. Indeed, in most cases if such a situation appears, then we could apply Lemma 9.19 to reduce the graph.

9.3.1 Proofs of Lemma 9.21 and Lemma 9.22

Color Game. Before proving Lemma 9.21 and Lemma 9.22, let us first introduce the color game. The *Color Game* has as input a set X of colors, p boxes and a set X_i of forbidden colors for each box. We denote by d_i the number of colors in X_i . We assume for simplicity that the number of forbidden colors is non-decreasing, *i.e.* $(d_i)_{1 \leq i \leq p}$ is a non-decreasing sequence. The goal of the game is to find the number of sets of p colors which can be one-to-one affected to the p boxes in such a way the color of the i -th box

is not a color of X_i . Such an assignment is called a *correct assignment* and the set of size p is a *correct set*. The Color Game is a function that assigns to every element of $(|X|, p, (X_i)_{i \leq p})$ an integer, the number of corresponding correct sets, which is called the *box choice*. Note that even if there are several correct assignments for the same set of colors, it still counts as only one. Let us prove two claims which are at the core of the proofs of Lemma 9.21 and Lemma 9.22.

Claim 9.26. *If $|X| = p$ and $d_i \leq i - 1$, then the box choice equals one.*

Proof. Clearly there is only one possible set (namely X), so the only thing to prove is that X is a correct set. We proceed by induction on p . Since $|X_p| \leq p - 1$ and $|X| = p$, there is a compatible color a for the p -th box. We forget the color a and the last box. There remain $(p - 1)$ boxes and $(p - 1)$ colors and d_i is non-increasing, so we apply induction on the remaining boxes. \square

We can actually generalize the claim when there is no restriction on the size of X .

Claim 9.27. *The box choice is at least $\binom{|X| - \max_i(d_i - i + 1)}{p}$.*

Proof. Let $\ell = |X|$. We proceed by induction on $\sum_{i=1}^p d_i$. If $\sum_{i=1}^p d_i = 0$, then every d_i equals 0. Since there is no constraint, all the subsets of size p are correct. So the box choice equals $\binom{\ell}{p}$.

So we may assume that $d_p \geq 1$ since d_i is non decreasing. If there exists an x such that x is in every X_i , then x is never in a correct assignment. Thus the box choice is not modified when x is deleted from X and from every X_i . Hence by induction, the box choice is at least $\binom{\ell - 1 - (\max_i(d_i - 1 - i + 1))}{p} = \binom{\ell - (\max_i(d_i - i + 1))}{p}$. In the following, we assume that every $x \in X$ does not appear in at least one X_i .

Let x be an element of X_p . Let j_x be the greatest integer such that $x \notin X_{j_x}$. The box choice is at least the number of correct sets which do not contain x plus the number of correct sets where x is assigned to the j_x -th box.

In the first case, let $d'_i = |X_i \setminus \{x\}|$ be the new number of forbidden colors of X_i after removing x , note that $d'_i \leq d_i$. In particular $d'_p = d_p - 1$ so we can apply the induction hypothesis and the box choice is at least $\binom{\ell - 1 - \max_i(d'_i - i + 1)}{p} \geq \binom{\ell - 1 - \max_i(d_i - i + 1)}{p}$, as $\binom{a}{p} \geq \binom{b}{p}$ when $a \geq b$.

In the second case, there remain $(p - 1)$ boxes to affect. Let d'_i be the new number of forbidden colors of the i th box after removing x and the box j_x . Let us prove $\max_i(d'_i - i) \leq \max_i(d_i - i)$. For $i < j_x$, the indices are not modified and $d'_i \leq d_i$. For $i > j_x$, the indices decrease by 1 (since the j_x -box is deleted) and $x \in X_i$ by definition of j_x , therefore $d'_{i-1} = d_i - 1$. In each case, $\max_i(d'_i - i) \leq \max_i(d_i - i)$. By induction, the box choice is at least $\binom{\ell - 1 - \max_i(d'_i - i + 1)}{p-1} \geq \binom{\ell - 1 - \max_i(d_i - i + 1)}{p-1}$.

Pascal's triangle ensures that the sum is at least:

$$\binom{\ell - 1 - \max_i(d_i - i + 1)}{p} + \binom{\ell - 1 - \max_i(d_i - i + 1)}{p - 1} = \binom{\ell - \max_i(d_i - i + 1)}{p}.$$

\square

We can now prove both lemmas.

Proof of Lemma 9.21. Let u be a vertex of G adjacent to p vertices v_1, \dots, v_p of degree different from $d(u)$. Let $c(v_i)$ be the number of neighbors of v_i that are of degree $d(v_i)$ and $m = \max_i(d(v_i) + c(v_i) - i)$. Assume in addition that $\binom{k+1-(d(u)-p)-m}{p} \geq d_{d(u)}(u) + 1$.

Let G' be the graph $G - \{uv_i\}_i$. Note that $G' \prec_P G$ and $G' \prec_M G$ and G' is still connected by Lemma 9.19. Also remark that $\text{mad}(G') \leq \text{mad}(G)$ and G' is planar if G is. Let α be the given AVD $(k + 1)$ -coloring of G' where u can possibly have conflicts with other vertices.

Let X be the subset of colors which are not adjacent to u in G' . We have $|X| = k + 1 - (d(u) - p)$. For every v_i , the set X_i of colors of X which cannot be given to uv_i is the set of colors adjacent to v_i (at most $d(v_i) - 1$ colors) plus the set of colors which create a conflict with a neighbor of v_i (at most $c(v_i)$ colors). So the set X_i has size at most $d(v_i) + c(v_i) - 1$. Note that a correct assignment of the color game on this instance is a coloring which does not yield any conflict for v_i .

Claim 9.27 ensures that at least $d_{d(u)}(u) + 1$ sets do not create any conflict for the vertices v_i . Since the total number of possible conflicts for u is $d_{d(u)}(u)$ and u cannot be in conflict with a v_i , at least one coloring does not create a conflict for u . It provides an AVD $(k + 1)$ -coloring of G .

In particular, if G' is AVD $(k + 1)$ -colorable, then G also is. Thus G is reducible. \square

Proof of Lemma 9.22. Assume that G contains a vertex u adjacent to p vertices v_1, \dots, v_p with degrees different from $d(u)$ such that $d(v_p) \geq 2$ and $\max_i(d(v_i) + c(v_i) - i) = 0$. Let G' be the graph obtained from G by deleting the edges uv_i for every i and adding p vertices w_1, \dots, w_p of degree one adjacent to u . Note that $G' \prec_P G$ as n_{2+} is non-increasing, the number of edges is constant and n_1 increases since $d(v_p) \geq 2$. Also $G' \prec_M G$ since $d(v_p) \geq 2$ decreases and no degree increases.

Assume that G' is AVD $(k + 1)$ -colorable and let α be such a coloring. Let us prove that we can extend α to G by coloring the edges uv_1, \dots, uv_p with colors $X = \{\alpha(uw_1), \dots, \alpha(uw_p)\}$. We denote by X_i the set of colors adjacent to v_i plus the set of colors which create a conflict with v_i and denote by d_i the size of X_i . Note that $d_i - i = d(v_i) - 1 + c(v_i) - i \leq -1$. By Claim 9.26, there exists a correct assignment for which does not yield a conflict for v_i . Moreover, there is no conflict for u since the set of colors adjacent to u in G and G' are the same. \square

9.3.2 Back to other lemmas

Let us prove Lemma 9.23 and Lemma 9.24.

Proof of Lemma 9.23.1. Assume that there is a vertex u of degree at least 2 adjacent to a vertex v_1 of degree 1 and to a vertex v_2 of degree 2. By Lemma 9.20, we can assume that u is of degree at least 3. Recall that $c(v_i)$ is the number of neighbors of v_i that are of the same degree, so we have $c(v_1) = 0$.

Suppose first that $c(v_2) \geq 1$, as v_2 has only one neighbor different from u , $c(v_2) = 1$. Let x be the neighbor of v_2 of degree 2 and w its other neighbor different from v_2 . We color $G - xv_2$ by minimality. If uv_2 and xw have the same color, we can permute the colors of uv_2 and uv_1 . Thus we can suppose that they have distinct colors. In this case x and v_2 have different set of colors and there is at least one free color for xv_2 which does not create any conflict as $k \geq 4$. Thus the graph is reducible. If $c(v_2) = 0$, then

$\max_i(d(v_i) + c(v_i) - i) = 0$ and v_1 and v_2 are not of the same degree as u . Thus, by Lemma 9.22, the graph is reducible. \square

Proof of Lemma 9.23.2. Indeed, assume that there is a vertex u of degree at least 2 adjacent to a vertex v_1 of degree 1 and to no more than $\lfloor \frac{k}{2} \rfloor$ neighbors of degree at least 3. By Lemma 9.20, we can assume that u is of degree at least three. Let v_2, \dots, v_p be the other neighbors of u of degree at most 2 (if any). By Lemma 9.23.1, all of them are of degree one. So $\max_i(d(v_i) + c(v_i) - i) = 0$. Since $d(u) - p \leq \lfloor \frac{k}{2} \rfloor$ and $p \neq k + 1 - (d(u) - p)$, we have $\binom{k+1-(d(u)-p)}{p} \geq k + 1 - (d(u) - p) \geq \lfloor \frac{k}{2} \rfloor + 1 \geq d_{d(u)}(u) + 1$. Thus, since $d(u) \geq 3$, by Lemma 9.21, the graph is reducible. \square

Let uv and uw be two edges. We *switch* the colors of uv and uw when we color uw with the color of uv and conversely. Note that if there is no conflict for u before switching the colors, there is no conflict for u after the switch since the set of colors adjacent to u is not modified.

Proof of Lemma 9.24. The proof is heavily inspired from a beautiful proof by Woodall [199] for a similar statement in the case of list edge-coloring with Δ colors.

Let $i < \lfloor \frac{k}{2} \rfloor$. Let D_i be the set of vertices of G of degree at most i and at least 2. Let B be the set of vertices with degree at least $\lfloor \frac{k}{2} \rfloor + 1$. Let B_i be the subset of vertices of B adjacent to at least one vertex of D_i . The connected components of the graph induced by D_i are singletons or edges by Lemma 9.20. Moreover, vertices of D_i are adjacent to either other vertices in D_i or vertices in B_i by Lemma 9.20. Consider the multigraph obtained from G by identifying the vertices of D_i which are adjacent. Denote by A_i the image of the set D_i when edges are contracted, and consider the bipartite multigraph G_i on vertex set (A_i, B_i) . Note that we keep multi-edges and delete the edges inside B_i . Every vertex of A_i has degree at most $2(i - 1)$ and $|D_i| \leq 2|A_i|$. By the hypothesis of Lemma 9.24, $|A_i| > 2(i - 1)|B_i|$.

We start with an easy structural claim on bipartite multigraphs.

Claim 9.28. *Let H be a bipartite multigraph with vertex set $V(H)$ bipartitioned into $A \cup B$, with $B \neq \emptyset$. For $\alpha > 0$, if for every non-empty subset $A' \subseteq A$ and $B' = N(A') \subseteq B$, there exists a vertex $u \in B'$ with $d_{A'}(u) < \alpha$, then $\alpha|B| > |A|$.*

Proof. By induction on $|A|$. If $|A| < \alpha$, since $|B| \geq 1$, the conclusion holds. If $|A| \geq \alpha$, there exists $u \in B$ with $d(u) < \alpha$. We apply the induction hypothesis to the graph $H - (N(u) \cup \{u\})$. It follows that $\alpha(|B| - 1) > |A| - \alpha$, hence the result. \blacksquare

We claim that there is a non-empty set $S \subseteq A_i$ such that the subgraph H of G_i obtained by considering $G_i[S \cup N(S)]$ is such that every vertex in $B' = N(S)$ has degree at least $2(i - 1)$ in H . Indeed, otherwise, in every non-empty subset $S \subseteq A_i$ and $B' = N(S) \subseteq B_i$, there exists a vertex $u \in B'$ with $d_H(u) < 2(i - 1)$. Thus Claim 9.28 holds and $2(i - 1)|B_i| > |A_i|$, a contradiction.

Consequently, there is a non-empty set $S \subseteq A_i$ such that the subgraph H of G_i obtained by considering $G_i[S \cup N(S)]$ is such that every vertex in $B' = N(S)$ has degree at least $2(i - 1)$ in H . Let S' be the pre-image of S before the contraction of edges in D_i . As S' is a non-empty set of vertices of degree at least 2, we can color by minimality of $G \otimes S'$.

To each vertex $u \in B'$, assign a list of $d_{S'}(u)$ colors corresponding to the color of each edge incident with u and to a vertex in S' . Similarly, this corresponds to an edge list

assignment of H such that $\forall u \in B', \forall v \in S$, if uv is an edge, then $|L(uv)| = d_H(u) \geq 2(i-1) \geq d_H(v)$.

To obtain our AVD $(k+1)$ -coloring of G , we use the following result.

Theorem 9.29. [24] *Let G be a bipartite multigraph and suppose that every edge $e = uv$ of G is given a list $L(e)$ of at least $\max\{d(u), d(v)\}$ colors. Then G is edge- L -colorable.*

Consequently, Theorem 9.29 applies, and H can be colored. To obtain an AVD $(k+1)$ -coloring of G , it suffices to color all edges which are in $G \otimes S'$ and not in H by their color in the coloring of $G \otimes S'$. All edges in H are colored by their color in H and it remains to color every uncolored edge between two vertices of S' . Note that at this moment, all vertices which are not in S' have the same set of colors as in the AVD $(k+1)$ -coloring of $G \otimes S'$. Two adjacent vertices of S' are already distinguished since they were contracted in H , so their incident sets of colors will be disjoint except for the color of their common edge. To color an edge between vertices of S' , it suffices to take any color beside the $2(i-1) \leq k$ that may already appear around one of the two vertices. Note that this is an AVD $(k+1)$ -coloring. Hence the conclusion, G is reducible. \square

9.3.3 Proof of Lemma 9.25

We will prove all configurations of Lemma 9.25. When proving that a configuration is P -reducible, we suppose that G does not contains any of the previous configurations.

Proof of Lemma 9.25.1. Assume there are two adjacent vertices v_1 and v_2 of degree 2. Let w_1 (resp. w_2) be the neighbor of v_1 (resp. v_2) other than v_2 (resp. v_1). By Lemma 9.20.2, $d(w_1), d(w_2) \neq 2$. Depending on whether $w_1 = w_2$, we color by minimality $G - v_1v_2$ or the graph obtained from G by contracting the edge v_1v_2 . Note that the resulting graph is still planar. In both cases, the edges w_1v_1 and w_2v_2 receive two different colors since they are adjacent. Finally since $d(w_i) \neq 2$, any coloring of v_1v_2 with a third color induces an AVD $(k+1)$ -coloring of G . \square

Proof of Lemma 9.25.2. Suppose that u is adjacent to two vertices v_1 and v_2 of degree 2. Let w_1 (resp. w_2) be the neighbor of v_1 (resp. v_2) that is not u . By Lemma 9.25.1, $d(w_1), d(w_2) \neq 2$. We consider two different cases depending on whether u and w_1 (resp. w_2) are adjacent.

- Assume, up to symmetry, that u is adjacent to w_1 . Let α be an AVD coloring of $G' = G \otimes \{v_1\}$. If $\alpha(uv_1) \neq \alpha(v_1w_1)$, then the coloring can be immediately extended to G . Otherwise, if $\alpha(uv_1) \neq \alpha(v_2w_2)$, switch the colors uv_1 and uv_2 and apply the first case. Finally, in the last case, switch the colors of uw_1 with the color of uv_1 and v_1w_1 (which are the same otherwise we can apply the first point). After the switching, the second case holds since the color of uv_1 has been modified.
- Assume that u is adjacent neither to w_1 nor to w_2 . Let G' be the graph obtained by deleting v_1 and v_2 and adding the edges uw_1 and uw_2 . Let α be an AVD coloring of G' . We color uv_1 and v_2w_2 with $\alpha(uw_2)$, and uv_2 and v_1w_1 with $\alpha(uw_1)$. Vertices v_1 and v_2 cannot raise a conflict, and the set of colors incident with each other vertex has not changed, thus this produces an AVD $(k+1)$ -coloring of G .

\square

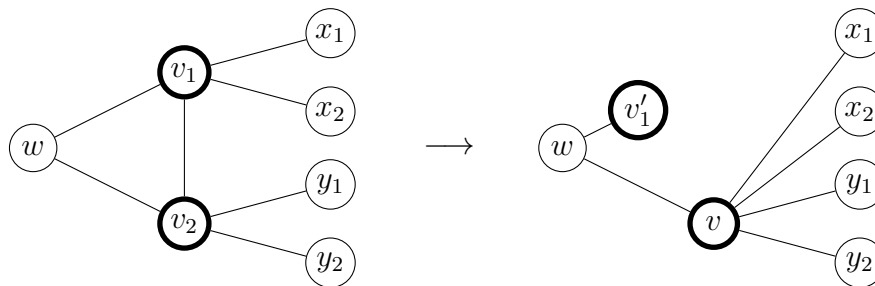


Figure 9.2: One case of Lemma 9.25.3. Here $d(v_1) = d(v_2) = 4$ and they have one common neighbor w . On the left is the neighborhood of v_1 and v_2 in G and on the right, the resulting graph G' . Vertices with bold boundary have no other edges in the graph.

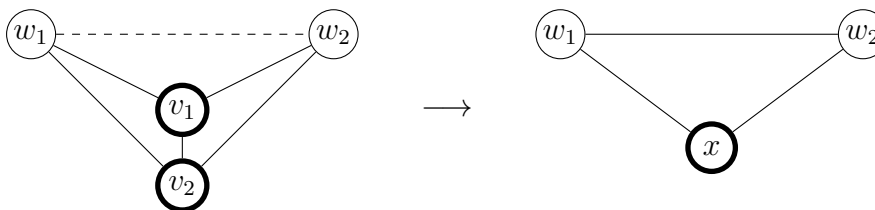


Figure 9.3: Illustration of Lemma 9.25.4. On the left is the neighborhood of v_1 and v_2 in G and on the right, the resulting graph G' . Vertices with bold boundary have no other edges in the graph. Dashed edges represent non-edges.

Proof of Lemma 9.25.3. Assume that G has two adjacent vertices v_1 and v_2 of degree $d \in \{3, 4, 5\}$ such that the edge v_1v_2 does not belong to two triangles (see an example on Figure 9.2). By Lemma 9.20.2, the only neighbor of v_1 (resp. v_2) of degree d is v_2 (resp. v_1). In particular, modifying the colors incident with v_1, v_2 will not create any conflict with other neighbors.

Since the edge v_1v_2 does not belong to two triangles, v_1 and v_2 have at most one neighbor in common. If v_1 and v_2 have exactly one neighbor w in common, we split v_1 into a vertex v'_1 of degree 1 adjacent to w and a vertex v''_1 of degree $d - 1$ with the remaining adjacencies. Then, in both cases, we contract the edge v_1v_2 (or v'_1v_2), thus creating a vertex v , and color by minimality the resulting graph G' . Indeed remark that, by construction, G' is planar, simple, of maximum degree at most k , and $G' \prec_p G$. We obtain a coloring of G by merely switching the colors of wv_1 and wv_2 if necessary to ensure a proper coloring, and by coloring v_1v_2 arbitrarily. Since v_1 and v_2 have no other neighbor of degree d , there is no other conflict to deal with. All their respective incident edges, except maybe one, were adjacent in the smaller graph. Hence, except wv'_1 , all edges incident with v have pairwise different colors. Since $d \geq 3$, this ensures that there is no conflict between v_1 and v_2 . The edge v_1v_2 has at most $2d - 2 \leq k$ constraints, so we can color it. \square

Proof of Lemma 9.25.4. Let v_1 and v_2 be the two adjacent 3-vertices. By Lemma 9.25.3, v_1 and v_2 belong to at least two common triangles. Let w_1 and w_2 their two common neighbors. By assumption, w_1 and w_2 are not adjacent (see See Figure 9.3). We construct G' from G by removing v_1 and v_2 , adding w_1w_2 , w_1x and xw_2 where x is a new vertex. The graph G' is planar and P -smaller than G . Let α be an AVD $(k + 1)$ -coloring of G' . Color v_1w_1 and v_2w_2 with color $\alpha(w_1w_2)$, v_1w_2 with color $\alpha(xw_2)$ and v_2w_1 with

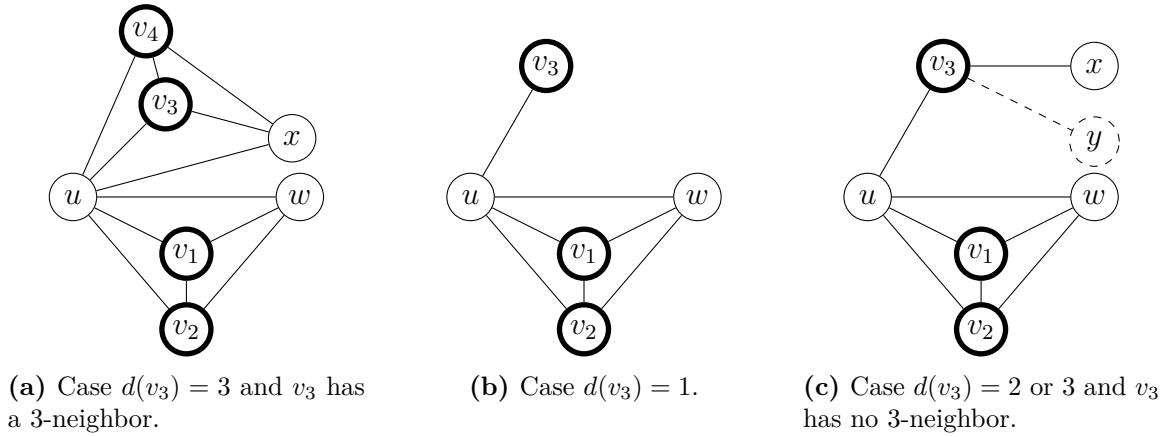


Figure 9.4: Illustration of Lemma 9.25.5. Vertices with bold boundary have no other edges in the graph. Dashed edges represent non-edges.

color $\alpha(xw_1)$. Finally color v_1v_2 properly. By construction $\alpha(xw_1) \neq \alpha(xw_2)$, thus v_1 and v_2 are not in conflict and we did not modify the color set of any other vertex. \square

Proof of Lemma 9.25.5. Suppose u is adjacent to two adjacent 3-vertices v_1 and v_2 and to a vertex v_3 of degree at most 3. Note that by Lemma 9.25.4, the graph G is such that v_1 and v_2 are both adjacent to a same vertex $w \neq u$ such that uw is an edge. By Lemma 9.20, $d(w) > 3$. We consider three cases (see Figure 9.4).

Let us first consider the case where v_3 also has a neighbor v_4 of degree 3 (hence $d(v_3) = 3$). Then by Lemma 9.25.4, the vertex v_4 is also adjacent to u , and both v_3 and v_4 are adjacent to a same vertex $x \neq u$ (which may or may not differ from w) such that ux is an edge. Again, we have $d(x) > 3$. In that case, we consider the graph $G' = G \otimes \{v_1, v_2, v_3, v_4\}$, and obtain an AVD $(k + 1)$ -coloring α of G' by minimality. We switch if necessary the colors of $\{uv_1, uv_2, uv_3, uv_4\}$ so that $\alpha(uv_1) \notin \{\alpha(wv_1), \alpha(wv_2)\}$ and $\alpha(uv_3) \notin \{\alpha(xv_3), \alpha(xv_4)\}$. This is possible as the set is of size 4. Then, by switching if necessary the colors of wv_1 and wv_2 , or those of xv_3 and xv_4 , and coloring arbitrarily the edges v_1v_2 and v_3v_4 , we obtain an AVD $(k + 1)$ -coloring of G .

Consider now the case where v_3 has no neighbor of degree 3. Suppose that v_3 has degree 1. Let G' be the graph obtained from G by deleting the edge v_1v_2 . We obtain an AVD $(k + 1)$ -coloring α of G' by minimality. If $\{\alpha(uv_1), \alpha(uv_2)\} \neq \{\alpha(wv_1), \alpha(wv_2)\}$, then we can add v_1v_2 back and color it arbitrarily. Otherwise, without loss of generality $\alpha(v_1w) \neq \alpha(uv_3)$. By switching the colors of uv_1 and uv_3 , the first case holds.

Now, v_3 is a 2-vertex or a 3-vertex. Let x (and y if it exists) be its two neighbors other than u . Let G' be the graph obtained from G by deleting the edge uv_3 and adding a pendant vertex v'_3 to u . We obtain an AVD $(k + 1)$ -coloring α of G' by minimality. We uncolor the edge v_1v_2 . Assume w.l.o.g. that $\alpha(uv_i) = i$ for $1 \leq i \leq 3$. Let 4 be the color of uw . If v_3 has no edge colored 3, other than uv_3 , then we can color uv_3 with 3. If none of wv_1 or wv_2 is colored 3, then by switching the colors of uv_3 with one of wv_1 or wv_2 we obtain the desired coloring after coloring v_1v_2 .

W.l.o.g. assume that wv_1 is colored 3. If $\alpha(uv_2) \notin \{1, 2, 3\}$, then we can extend the coloring by setting $\alpha(uv_3) = 2$ and $\alpha(uv_2) = 3$ or by setting $\alpha(uv_3) = 1$, $\alpha(uv_2) = 3$ and $\alpha(uv_1) = 2$, and then by coloring v_1v_2 properly. Now we can assume that $\alpha(wv_2) \in \{1, 2, 3\}$ and thus $\alpha(wv_2) = 1$. By coloring uv_3 and wv_1 with 1 (resp. 4), wv_1 and wv_2

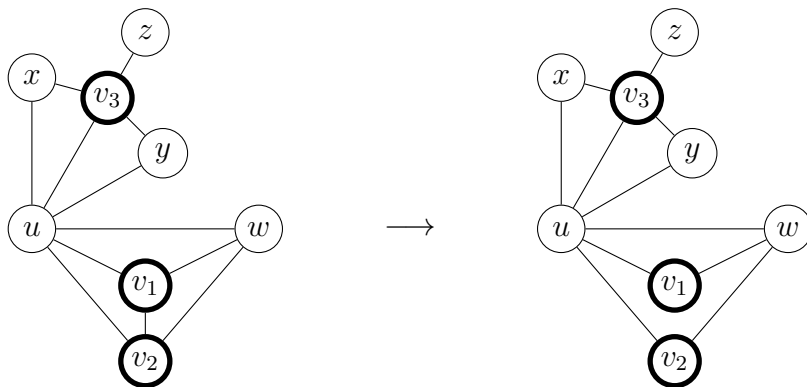


Figure 9.5: Illustration of Lemma 9.25.6. On the left is the neighborhood of u in G and on the right, how it is modified in G' . Vertices with bold boundary have no other edges in the graph.

with 4 (resp. 3) and uw with 3 (resp. 1), we obtain an AVD $(k + 1)$ -coloring of G when v_3 has no edge colored 1 (resp. 4), other than uv_3 . Note that v_3 cannot be incident with an edge colored 3, another colored 1 and a third colored 4 when all of these edges are different from uv_3 since v_3 has degree at most 3. \square

Proof of Lemma 9.25.6. Suppose u is adjacent to two adjacent 3-vertices v_1 and v_2 and to a weak vertex v_3 of degree 4 (see Figure 9.5). Note that by Lemma 9.25.4, the graph G is such that v_1 and v_2 are both adjacent to a same vertex $w \neq u$ such that uw is an edge. By Lemma 9.20, we have $d(w) > 3$.

Let x, y and z be the three other neighbors of v_3 . Let G' be the graph obtained from G by deleting the edge v_1v_2 . We obtain an AVD $(k + 1)$ -coloring α of G' by minimality. Assume w.l.o.g. that $\alpha(uv_i) = i$ for $1 \leq i \leq 3$ and $\alpha(uw) = 4$. If $\{\alpha(wv_1), \alpha(wv_2)\} \neq \{1, 2\}$, then we can add v_1v_2 back and color it arbitrarily.

If v_3 is adjacent to a 4-vertex, assume w.l.o.g. that it is z . Uncolor v_3z and let S be the set of colors containing $\alpha(v_3x), \alpha(v_3y)$ and possibly a color a which creates a conflict with z when uv_3 is colored a . If v_3 is not adjacent to a 4-vertex then let S be the set of colors containing $\alpha(v_3x), \alpha(v_3y)$ and $\alpha(v_3z)$. In both case, if we color uv_3 with a color not in S , then there will be no conflict for v_3 .

Assume $S \neq \{1, 2, 4\}$. If $1 \notin S$, then exchange the colors of uv_1 and uv_3 , recolor v_1v_2 and possibly v_3z . The same holds if $2 \notin S$. If $4 \notin S$, then color uw with 1 and the two edges uv_1 and uv_2 with 4 to get back to the previous case.

Assume now $S = \{1, 2, 4\}$. Note that if $d(z) = 4$, then S is the set of colors of the three colored edges of z . Without loss of generality, assume that uxv_3 is a triangular face of G . Let $b \in S$ be the color of v_3x . In particular $x \neq w$ since w is already incident with all three colors of S on wu, wv_1 and wv_2 . W.l.o.g. ux is colored 5. If $b = 1$, then color v_3x with 5, ux with 1 and uv_1 with 5. A similar construction works for $b = 2$. If $b = 4$, then color v_3x with 5, ux with 4, uv_1 with 5, v_2w with 2 and uw with 1. In both cases, we can recolor v_1v_2 (and possibly v_3z) arbitrarily without conflict. \square

Proof of Lemma 9.25.7. Suppose u is adjacent to two adjacent 3-vertices v_1 and v_2 and $4 \leq d(u) \leq 9$ (see Figure 9.6). Note that by Lemma 9.25.4, the graph G is such that v_1 and v_2 are both adjacent to a same vertex $w \neq u$ such that uw is an edge. Let G' be the graph obtained from G by deleting the edge v_1v_2 . We obtain an AVD $(k + 1)$ -coloring α of G' by

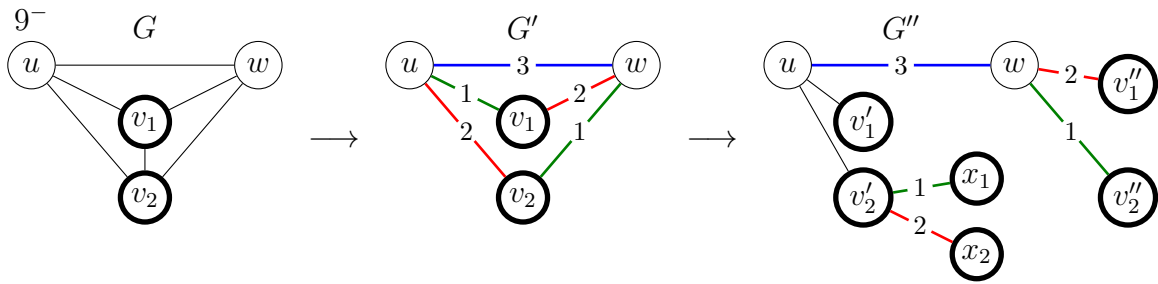


Figure 9.6: Illustration of Lemma 9.25.7. On the left is the neighborhood of u in G , in the middle is the graph G' with the coloring requiring the creation of the graph G'' which is on the right. Vertices with bold boundary have no other edges in the graph.

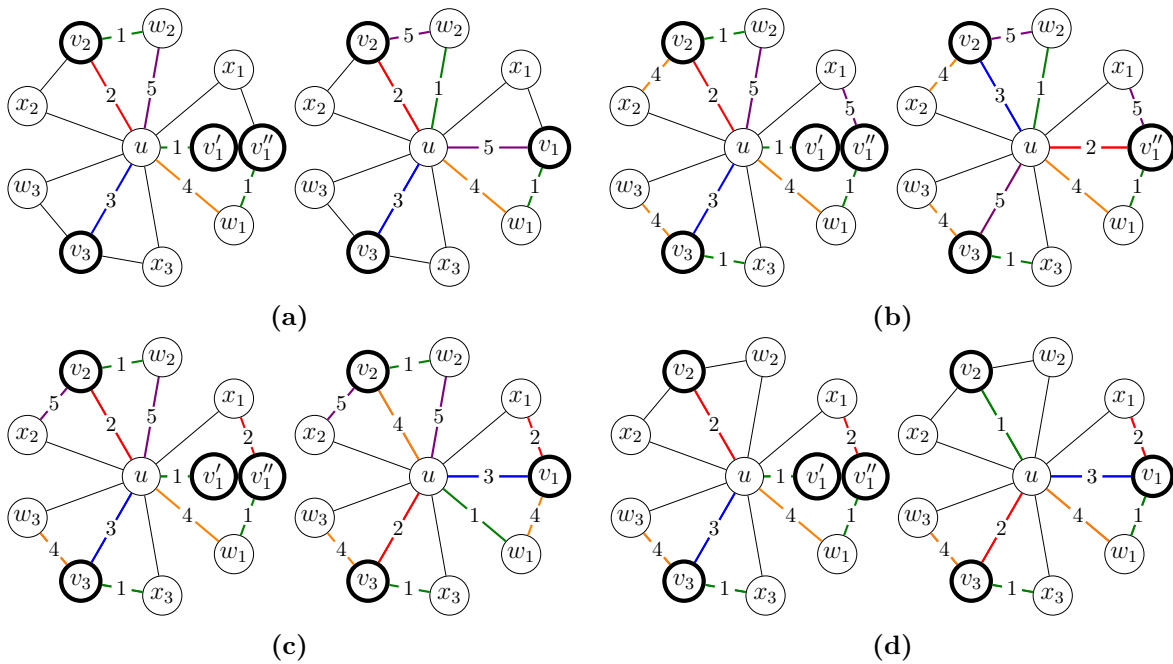


Figure 9.7: Illustration of Lemma 9.25.8 representing part of the neighborhood of u in G . Vertices with bold boundary have no other edges in the graph. Each subcase is composed of the initial coloring on G' on the left and the constructed coloring of G on the right.

minimality. Assume w.l.o.g. that $\alpha(uv_i) = i$ for $1 \leq i \leq 2$. If $\{\alpha(uv_1), \alpha(uv_2)\} \neq \{1, 2\}$, then we can add v_1v_2 back and we can color it arbitrarily.

Let G'' be the graph obtained from $G' \otimes \{v_1, v_2\}$ by adding two edges $v_2'x_1, v_2'x_2$ where v_2' is the neighbor of u obtained from splitting v_2 (see Figure 9.6). Note that $G'' \prec_P G$. We can extend α to $G'' - \{uv_1', uv_2'\}$ by coloring $v_2'x_1$ and $v_2'x_2$ with colors 1 and 2. We can apply Lemma 9.21 to color G'' . Indeed, $m = \max(1 - 1, 3 - 2) = 1$ and $p = 2$, thus $\binom{k+1-d(u)+p-m}{p} \geq \binom{5}{2} = 10 \geq d_{d(u)}(u) + 1$. Now let a and b be the colors of uv_1' and uv_2' in this new coloring. Color G with the colors of α and recolor uv_1 and uv_2 with a and b . As $b \neq 1, 2$, v_1 and v_2 do not have the same colors by the construction of the coloring in Lemma 9.21 and we can properly recolor v_1v_2 . \square

Proof of Lemma 9.25.8. Assume that G has a vertex u adjacent to v_1, v_2, v_3 of degree 3 where v_1, v_2 are weak neighbors of u (see Figure 9.7). By Lemma 9.25.5, $\{v_1, v_2, v_3\}$ is a stable set. For $1 \leq i \leq 3$, let w_i and x_i be the two other neighbors of v_i . Note that

Lemma 9.25.5 ensures that none of $w_1, w_2, w_3, x_1, x_2, x_3$ is of degree 3. However, some of the w_i 's and x_i 's may coincide.

Let G' be the graph obtained from G by deleting uw_1 and adding a vertex v'_1 adjacent only to u . We obtain an AVD $(k+1)$ -coloring α of G' by minimality. Consider w.l.o.g. that $\alpha(uv'_1) = 1$, $\alpha(uv_2) = 2$, and $\alpha(uv_3) = 3$.

If $1 \notin \{\alpha(v_1x_1), \alpha(v_1w_1)\}$, then we can extend α to G . Hence we can assume w.l.o.g. that $\alpha(v_1w_1) = 1$. Consider w.l.o.g. that $\alpha(uw_1) = 4$.

Similarly, for each $i \in \{2, 3\}$, if $1 \notin \{\alpha(v_ix_i), \alpha(v_iw_i)\}$ and $i \neq \alpha(v_1x_1)$, then we switch the colors of uv_i and uv_1 , thus obtaining an AVD $(k+1)$ -coloring of G . Therefore, for each $i \in \{2, 3\}$, it holds that $1 \in \{\alpha(v_ix_i), \alpha(v_iw_i)\}$ or $\alpha(v_1x_1) = i$.

If $\alpha(v_1x_1) \neq 4$, then we can color v_1w_1 and uv'_1 with 4 and uw_1 with 1, and we repeat the previous arguments. Therefore, if $\alpha(v_1x_1) \neq 4$, for each $i \in \{2, 3\}$, then it holds that $\{\alpha(v_ix_i), \alpha(v_iw_i)\} = \{1, 4\}$ or $\alpha(v_1x_1) = i$.

Assume first that $1 \in \{\alpha(v_2x_2), \alpha(v_2w_2)\}$. Assume w.l.o.g. $1 = \alpha(v_2w_2)$. Note that $w_1 \neq w_2$. Let $5 = \alpha(uw_2)$ and observe that $5 \notin \{1, 2, 3, 4\}$. We consider the following three cases.

- If $5 \notin \{\alpha(v_2x_2), \alpha(v_1x_1)\}$ (see Figure 9.7.a), then we color w_2v_2 with 5, uw_2 with 1, uw_1 with 5: this provides an AVD $(k+1)$ -coloring of G .
- If $5 = \alpha(v_1x_1)$ (see Figure 9.7.b), then, in particular $\alpha(v_1x_1) \notin \{2, 3, 4\}$ and $\{\alpha(v_2x_2), \alpha(v_2w_2)\} = \{\alpha(v_3x_3), \alpha(v_3w_3)\} = \{1, 4\}$. We can easily extend the coloring to an AVD $(k+1)$ -coloring of G by setting $\alpha(w_2v_2) = 5$, $\alpha(uw_2) = 1$, $\alpha(uv_2) = 3$, $\alpha(uv_1) = 2$ and $\alpha(uv_3) = 5$.
- If $5 = \alpha(v_2x_2)$ (see Figure 9.7.c), then this implies $\alpha(v_1x_1) = 2$ and $\{\alpha(v_3x_3), \alpha(v_3w_3)\} = \{1, 4\}$. We can obtain an AVD $(k+1)$ -coloring of G by setting $\alpha(uw_1) = 1$, $\alpha(w_1v_1) = 4$, $\alpha(uv_1) = 3$, $\alpha(uv_2) = 4$ and $\alpha(uv_3) = 2$.

Assume now $1 \notin \{\alpha(v_2x_2), \alpha(v_2w_2)\}$ (see Figure 9.7.d). This implies $\alpha(v_1x_1) = 2$, which in turn implies $\{\alpha(v_3x_3), \alpha(v_3w_3)\} = \{1, 4\}$. We can extend the coloring to an AVD $(k+1)$ -coloring of G by setting $\alpha(uv_1) = 3$, $\alpha(uv_2) = 1$ and $\alpha(uv_3) = 2$. \square

Before proving Lemma 9.25.9, Lemma 9.25.10 and Lemma 9.25.11, we present a generic reduction technique. Suppose u is adjacent to two adjacent weak 4-vertices v_1 and v_2 . Let x_1 and x_2 such that ux_1v_1 and ux_2v_2 are triangular faces of G . By Lemma 9.25.3, v_1 and v_2 have a common neighbor w different from u . When $w \in \{x_1, x_2\}$, we say that we are in a type A configuration and in a type B configuration otherwise (see Figure 9.8). We construct the graph G' from G by identifying v_1 and v_2 to form a vertex v and by adding two 1-vertices t_1 and t_2 which are respectively adjacent to u and w .

We obtain an AVD $(k+1)$ -coloring α of G' by minimality. When we are in a type A configuration, suppose w.l.o.g. $w = x_1$, $\alpha(uv) = 1$, $\alpha(zv) = 2$, $\alpha(uw) = 3$ and $\alpha(x_2v) = 4$ where z is the fourth neighbor of v_1 (see Figure 9.8). When we are in a type B configuration, suppose w.l.o.g. $\alpha(uv) = 1$, $\alpha(x_1v) = 2$, $\alpha(uw) = 3$ and $\alpha(x_2v) = 4$ (see Figure 9.8).

Claim 9.30. *The following statements hold.*

1. *Suppose we are in a type A configuration.*

- (a) *If G' admits an AVD $(k+1)$ -coloring α such that $\alpha(ut_1) \neq \alpha(wt_2)$, then we can extend the coloring α to G .*

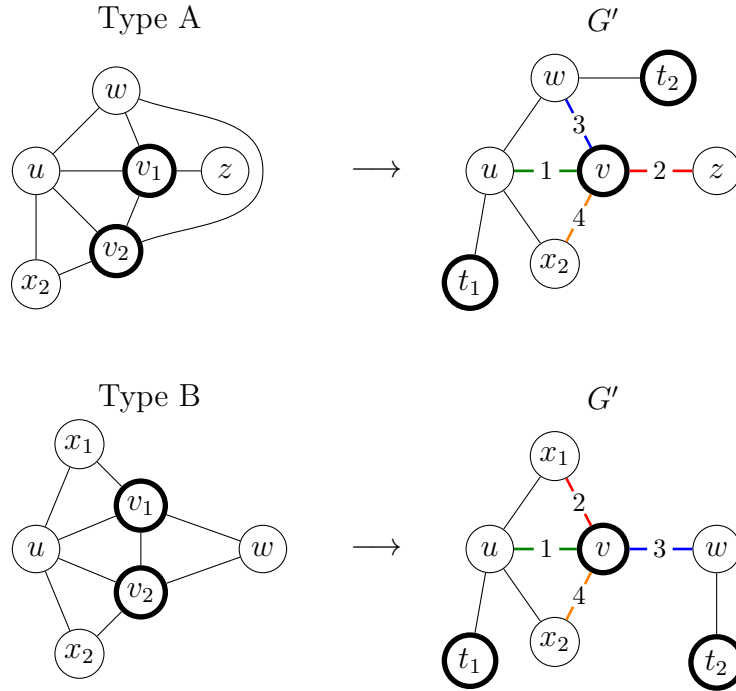


Figure 9.8: The two types of adjacent weak 4-neighbors of u . On the left is the neighborhood of v_1 and v_2 in G and on the right is the reduced graph G' .

- (b) If G' admits an AVD $(k + 1)$ -coloring α such that $\alpha(ut_1) = \alpha(wt_2) = c_1$, $\alpha(uw) = c_2$ and $\{c_1, c_2\} \neq \{2, 4\}$, then we can extend the coloring α to G .
- (c) If G' admits an AVD $(k + 1)$ -coloring α such that $\alpha(ut_1) = \alpha(wt_2) = c_1$, $\alpha(uw) = c_2$ and $\{c_1, c_2\} = \{2, 4\}$, then there exists an AVD $(k + 1)$ -coloring α' of G' such that $\alpha'(ut_1) = \alpha'(wt_2) = c_2$, $\alpha(uw) = c_1$, α and α' differ only on these three edges and every vertex r except t_1 and t_2 verify $\mathcal{C}_r(\alpha) = \mathcal{C}_r(\alpha')$.

2. Suppose we are in a type B configuration.

- (a) If G' admits an AVD $(k + 1)$ -coloring α such that either $\alpha(ut_1) \neq \alpha(wt_2)$ or $\alpha(ut_1) = \alpha(wt_2) \notin \{2, 4\}$, then we can extend the coloring α to G .
- (b) If G' admits an AVD $(k + 1)$ -coloring α such that $\alpha(ut_1) = \alpha(wt_2) \in \{2, 4\}$, say w.l.o.g $\alpha(ut_1) = 2$, and $\alpha(ux_1) \notin \{3, 4\}$ or, $\alpha(ux_1) = 4$ and $\alpha(ux_2) \neq 3$, then we can extend the coloring α to G .
- (c) If G' admits an AVD $(k + 1)$ -coloring α such that $\alpha(ut_1) = \alpha(wt_2) \in \{2, 4\}$, say w.l.o.g $\alpha(ut_1) = 2$, and $\alpha(ux_1) = 3$ or, $\alpha(ux_1) = 4$ and $\alpha(ux_2) = 3$, then there exists an AVD $(k + 1)$ -coloring α' of G' such that $\alpha'(ut_1) = \alpha'(wt_2) = 3$, α and α' only differ on edges among ux_1, ux_2 and the edges incident with v, t_1, t_2 . Moreover every vertex r except v, t_1 and t_2 verify $\mathcal{C}_r(\alpha) = \mathcal{C}_r(\alpha')$.

In other words, for both types, we can either directly extend α to G or we can modify α such that the color of ut_1 is different from its previous color. Remark that if $\alpha(ut_1) \neq \alpha(wt_2)$, then we can color G for both types.

Proof. Let us start with the proof of (1a), (1b), (2a) and (2b). In each case, we extend α to G . First, let $\alpha(ut_1) = a$ and $\alpha(wt_2) = b$. We want to prove (1a), hence we suppose $a \neq b$. See Figure 9.9 for an illustration of the following three cases.

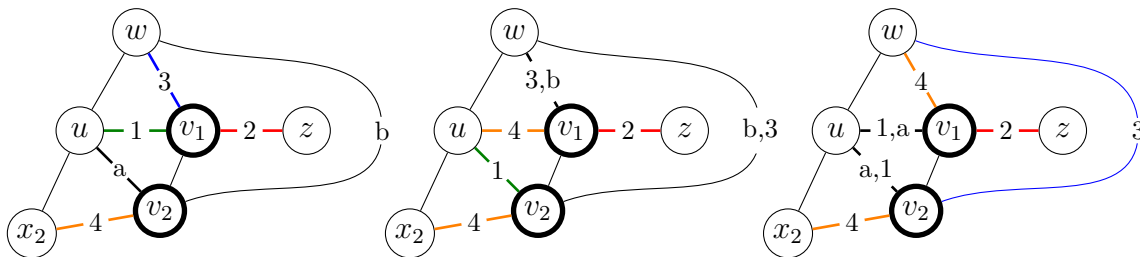


Figure 9.9: The coloring of G in each of the three cases of Claim 9.30.1a. On the left there is the case $a, b \neq 4$, in the middle it is the case $a = 4$ and on the right we have the case $b = 4$.

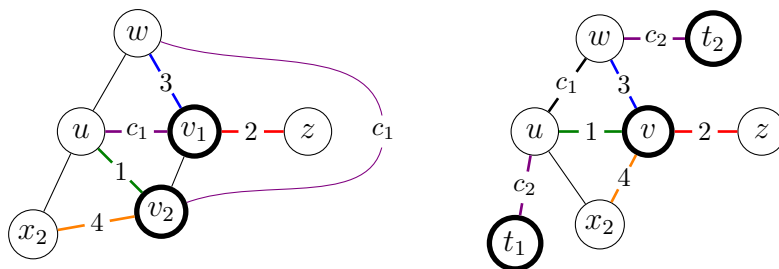


Figure 9.10: The two cases of Claim 9.30.1b. On the left there is the coloring of G for the case $c_1 \notin \{2, 4\}$, and on the right there is the coloring of G' for the case $c_1 \in \{2, 4\}$ and $c_2 \notin \{2, 4\}$.

- If both a and b are different from 4, then set $\alpha(uv_1) = 1$, $\alpha(zv_1) = 2$, $\alpha(wv_1) = 3$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = a$ and $\alpha(wv_2) = b$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly.
- If $a = 4$, then set $\alpha(uv_1) = 4$, $\alpha(zv_1) = 2$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = 1$. If $b \neq 1$, then set $\alpha(wv_2) = b$ and $\alpha(wv_1) = 3$, otherwise set $\alpha(wv_2) = 3$ and $\alpha(wv_1) = b = 1$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly.
- If $b = 4$, then set $\alpha(wv_1) = 4$, $\alpha(zv_1) = 2$, $\alpha(x_2v_2) = 4$, $\alpha(wv_2) = 3$. If $a \neq 3$, then set $\alpha(uv_2) = a$ and $\alpha(uv_1) = 1$, otherwise set $\alpha(uv_2) = 1$ and $\alpha(uv_1) = a = 3$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly.

This proves (1a). We want to prove (1b), hence we suppose $a = b = c_1$. Let $c_2 = \alpha(uw)$. See Figure 9.10 for an illustration of the following two cases.

- If $c_1 \notin \{2, 4\}$, then set $\alpha(uv_1) = c_1$, $\alpha(zv_1) = 2$, $\alpha(wv_1) = 3$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = 1$ and $\alpha(wv_2) = c_1$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly.
- If $c_1 \in \{2, 4\}$, then $c_2 \notin \{2, 4\}$ and take a coloring α' coinciding with α except for the following three edges: $\alpha'(ut_1) = \alpha'(wt_2) = c_2$ and $\alpha'(uw) = c_1$. Remark that it is an AVD $(k + 1)$ -coloring of G' for which we can apply the previous case.

This proves (1b). Remark that (1c) follows directly from the construction of the previous case when $\{c_1, c_2\} = \{2, 4\}$.

Let us continue with type B. We want to prove (2a). See Figure 9.11 for an illustration of the following three cases.

- If $a \neq b$ and both are different from 4, then set $\alpha(uv_1) = 1$, $\alpha(x_1v_1) = 2$, $\alpha(wv_1) = 3$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = a$ and $\alpha(wv_2) = b$. Remark that v_1 and v_2 cannot be in

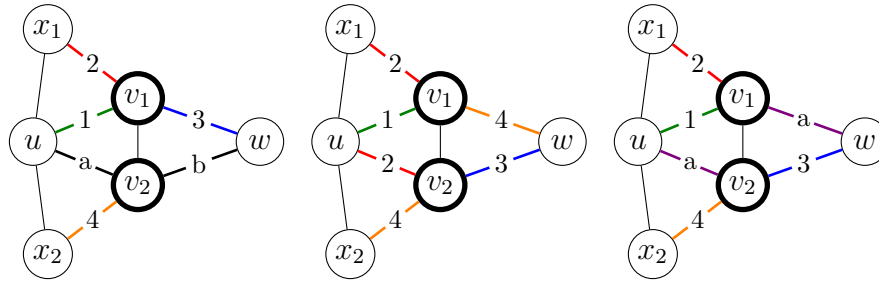


Figure 9.11: The coloring of G in each of the three cases of Claim 9.30.2a. On the left there is the case $a \neq b$ and $a, b \neq 4$, in the middle it is the case $a = 2$ and $b = 4$ and on the right we have the case $a = b \neq 2, 4$.

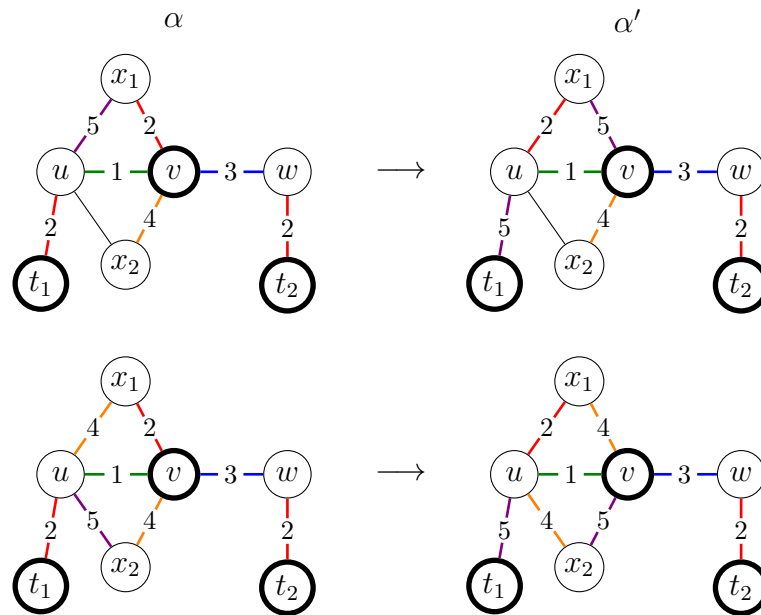


Figure 9.12: The two cases of Claim 9.30.2b. On the top, there is the case $\alpha(ux_1) = 5 \notin \{3, 4\}$, and on the bottom we have the case $\alpha(ux_1) = 4$ and $\alpha(ux_2) = 5 \neq 3$.

conflict and finally we can color (v_1v_2) properly. The case $a \neq b$ and $a, b \neq 2$ can be solved in a similar fashion.

- If $a \neq b$ and $a = 2$ and $b = 4$, then set $\alpha(uv_1) = 1$, $\alpha(x_1v_1) = 2$, $\alpha(wv_1) = 4$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = 2$ and $\alpha(wv_2) = 3$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly. The case $a \neq b$ and $a = 4$ and $b = 2$ can be solved in a similar fashion.
- If $a = b$ and both are different from 2 and 4, then set $\alpha(uv_1) = 1$, $\alpha(x_1v_1) = 2$, $\alpha(wv_1) = a$, $\alpha(x_2v_2) = 4$, $\alpha(uv_2) = a$ and $\alpha(wv_2) = 3$. Remark that v_1 and v_2 cannot be in conflict and finally we can color v_1v_2 properly.

This proves (2a). We now want to prove (2b), hence we suppose w.l.o.g $a = b = 2$. See Figure 9.12 for an illustration of the following two cases.

- If $\alpha(ux_1) \notin \{3, 4\}$, say $\alpha(ux_1) = 5$, then take a coloring α' coinciding with α except for the following three edges: $\alpha'(ut_1) = \alpha'(x_1v) = 5$, $\alpha'(ux_1) = 2$. The coloring α' is an AVD $(k + 1)$ -coloring for which $\alpha'(ut_1) \neq \alpha'(wt_2)$ and thus (2a) applies.

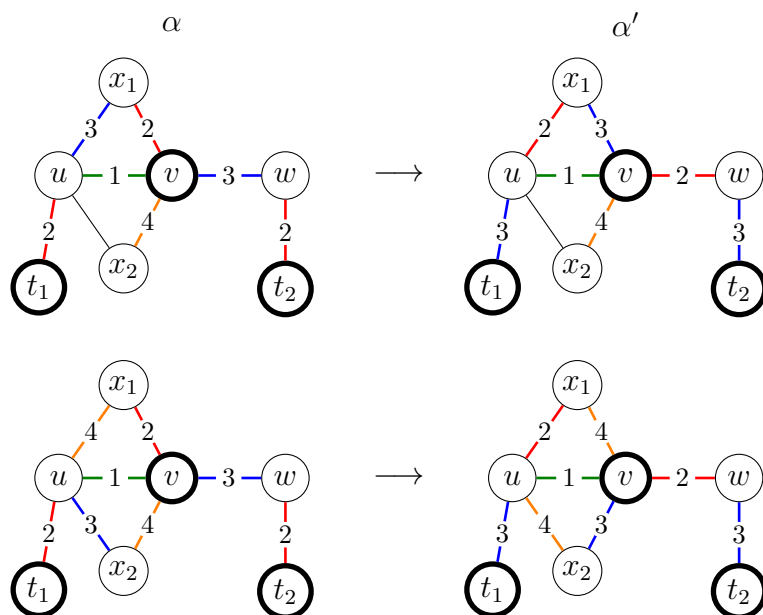


Figure 9.13: The two cases of Claim 9.30.2c. On the top, there is the case $\alpha(ux_1) = 3$, and on the bottom we have the case $\alpha(ux_1) = 4$ and $\alpha(ux_2) = 3$.

- If $\alpha(ux_1) = 4$ and $\alpha(ux_2) \neq 3$, say $\alpha(ux_2) = 5$, then take a coloring α' coinciding with α except for the following edges: $\alpha'(ut_1) = \alpha'(x_2v) = 5$, $\alpha'(ux_2) = \alpha'(vx_1) = 4$ and $\alpha'(ux_1) = 2$. The coloring α' is an AVD $(k + 1)$ -coloring for which $\alpha'(ut_1) \neq \alpha'(wt_2)$ and thus (2a) applies.

This proves (2b). We finally want to prove (2c), hence we suppose w.l.o.g $a = b = 2$. See Figure 9.13 for an illustration of the following two cases.

- If $\alpha(ux_1) = 3$, then take a coloring α' coinciding with α except for the following edges: $\alpha'(ut_1) = \alpha'(x_1v) = \alpha'(wt_2) = 3$ and $\alpha'(ux_1) = \alpha'(vw) = 2$. The coloring α' has the desired properties.
- If $\alpha(ux_1) = 4$ and $\alpha(ux_2) = 3$, then take a coloring α' coinciding with α except for the following edges: $\alpha'(ut_1) = \alpha'(x_2v) = \alpha'(wt_2) = 3$, $\alpha'(ux_1) = \alpha'(vw) = 2$ and $\alpha'(vx_1) = \alpha'(ux_2) = 4$. The coloring α' has the desired properties.

This proves (2c). □

We can now present the proofs of the remaining cases of Lemma 9.25.

Proof of Lemma 9.25.9. Suppose u is adjacent to two adjacent weak 4-vertices v_1 and v_2 and to a 2^- -vertex v_3 . Perform the generic reduction presented before Claim 9.30 to create G' from G . If we can apply Claim 9.30 to color G , then we have nothing to do. Otherwise, let a be the color of ut_1 . If v_3 is not incident with an edge colored a , then we can exchange the color of uv_3 and ut_1 . Now ut_1 and wt_2 do not have the same color, thus we can color G using Claim 9.30. If v_3 is incident with an edge colored a , then we can apply Claim 9.30 (1c or 2c) to change the color of ut_1 and then we apply the previous case. □

Proof of Lemma 9.25.10. Suppose u is adjacent to two adjacent weak 4-vertices v_1 and v_2 and to two adjacent weak 4-vertices v'_1 and v'_2 . Let w (resp. w') be the common neighbor

of v_1 and v_2 (resp. v'_1 and v'_2) other than u . Note that the two pairs of adjacent 4-vertices do not have to be of the same type (A or B).

Let G' be the graph obtained from G by contracting v_1v_2 to form a vertex v and by adding the two vertices t_1 and t_2 adjacent respectively to u and w . Let G'' be the graph obtained from G' by contracting $v'_1v'_2$ to form a vertex v' and by adding the two vertices t'_1 and t'_2 adjacent respectively to u and w' . Let α be an AVD $(k + 1)$ -coloring of G'' obtained by minimality.

If we can extend α to G by applying Claim 9.30 twice (from G'' to G' and then from G' to G), then we have nothing to do. Otherwise, we are in one of the non-extendable cases either on the side of v_1, v_2 or on the side of v'_1, v'_2 . Suppose w.l.o.g. that $\alpha(ut_1) = \alpha(wt_2)$. Let α' be the coloring obtained from α by exchanging the colors of ut_1 and ut'_1 . Now $\alpha'(ut_1) \neq \alpha'(wt_2)$. If we can extend α' to G using Claim 9.30, then we can conclude, otherwise it means that $\alpha'(ut'_1) = \alpha'(w't'_2)$. In particular $\alpha(wt_2) = \alpha(w't'_2)$. By applying Claim 9.30 to G'' colored with α , we can obtain α'' an AVD $(k + 1)$ -coloring of G'' where $\alpha''(w't'_2) = \alpha(w't'_2) = \alpha(wt_2) \neq \alpha''(wt_2)$. By applying the same procedure as before we are able to extend α'' to G since $\alpha''(w't'_2) \neq \alpha''(wt_2)$. \square

Proof of Lemma 9.25.11. Suppose u is a 9-vertex, with $d_9(u) \leq 4$, adjacent to two adjacent weak 4-vertices v_1 and v_2 . Let G' as presented in the context of Claim 9.30. Let α be an AVD $(k + 1)$ -coloring of G' . If we can apply Claim 9.30 to color G , then we are done.

If not, let us consider the graph G'' obtained from $G' \otimes v$ by deleting uv and adding uv' where v' is a new vertex. Also add an edge t_1y and three edges $v'x_i$ for $i \in \{1, 2, 3\}$. Consider the graph $G'' - \{ut_1, uv'\}$ and its coloring α' obtained from α by coloring t_1y by $\alpha(wt_2)$, and the three edges $v'x_i$ for $i \in \{1, 2, 3\}$, by the three colors $\{\alpha(vx_1), \alpha(vx_2), \alpha(vw)\}$. It is an AVD $(k + 1)$ -coloring of $G'' - \{ut_1, uv'\}$ except possibly for u . We can apply Lemma 9.21 to find an AVD $(k + 1)$ -coloring of G'' by coloring only ut_1 and uv' (note that G'' does not have to be P -smaller than G in order to apply Lemma 9.21). Indeed in this case, $p = 2$, $m = \max(2 - 1, 4 - 2) = 2$ and $\binom{12+1-9+p-m}{p} = \binom{4}{2} = 6 \geq 4 + 1$ by hypothesis.

Now we can color G' by identifying v and v' and removing the vertices y , x_1 , x_2 and x_3 . This is a proper AVD $(k + 1)$ -coloring of G' such that ut_1 and wt_2 have different colors. We can therefore apply Claim 9.30 to color G . \square

9.4 Proof of Lemma 9.20

This section is devoted to the proof of Lemma 9.20. The proof of this lemma is substantially more difficult than the proofs of the previous section.

9.4.1 Definitions

A partial edge-coloring γ is a *uv-partial edge-coloring* if all edges except uv are colored. Such a coloring is *safe* when no two vertices are in conflict for γ . We say that γ is *almost-safe* when no two vertices are in conflict for γ except possibly for u and v . In other words, a safe edge-coloring is an almost-safe edge-coloring where u and v are not in conflict.

Fix a *uv-partial safe edge-coloring* γ . We note $n_{inc}(\gamma) = |\mathcal{C}_{uv}(\gamma)|$ the number of incident colors with the edge uv . We note $\mathcal{C}^{com}(\gamma) = \mathcal{C}_u(\gamma) \cap \mathcal{C}_v(\gamma)$ the set of colors

which are both on an edge incident with u and an edge incident with v . The number $\varrho(\gamma) = |\mathcal{C}^{com}(\gamma)|$ is the number of such colors. Remark that when uv is the only uncolored edge, $n_{inc}(\gamma) = (d(u) - 1) + (d(v) - 1) - \varrho(\gamma)$.

A vertex $x \neq v$ is in a γ -conflict with u if it is in conflict with u when uv is colored with $c \notin \mathcal{C}_{uv}(\gamma)$ in γ . The set $S_u(\gamma)$ is the set of vertices in a γ -conflict with u and $Conf_u(\gamma)$ is the set of colors c for which u has a γ -conflict when uv is colored c . Let $F_u(\gamma) = |Conf_u(\gamma)|$ be the number of such colors, the inequality $F_u \leq |S_u|$ follows from the definitions. It is often more convenient to count separately the conflicts for u and for v .

We denote by $n_{compa}(\gamma)$ the number of colors for which uv can be properly colored without creating a conflict. Note that:

$$n_{compa}(\gamma) = k + 1 - n_{inc}(\gamma) - |Conf_u(\gamma) \cup Conf_v(\gamma)| \geq k + 1 - n_{inc}(\gamma) - F_u(\gamma) - F_v(\gamma).$$

9.4.2 Overview of the proof

The proof is organized as follows. We first state one quite technical lemma. The proof of this lemma will be detailed later. We then prove Lemma 9.20 which is a consequence of Lemma 9.31. This ensures that in a non-reducible graph, each vertex of degree at most $\lfloor \frac{k}{2} \rfloor$ is adjacent to at most one vertex of degree at most $\lfloor \frac{k-1}{2} \rfloor$, which must then have the same degree.

Lemma 9.31. *Let $G = (V, E)$ be a graph with $\Delta(G) \leq k$ which contains an edge uv with $d(u) \leq d(v) \leq \lfloor \frac{k}{2} \rfloor$ and $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$. Let $\delta = 1$ if $d(v) > \lfloor \frac{k-1}{2} \rfloor$ and $\delta = 0$ otherwise.*

If $d(u) < d(v)$ and there is a uv -partial safe $(k + 1)$ -edge-coloring γ of G , then there is a uv -partial safe $(k + 1)$ -edge-coloring α of G with $n_{compa}(\alpha) \geq 2 - \delta$.

If $d(u) = d(v)$ and there is a uv -partial safe $(k + 1)$ -edge-coloring γ of G , then there is a uv -partial safe $(k + 1)$ -edge-coloring α of G with $n_{compa}(\alpha) \geq 1$.

Let us now prove Lemma 9.20.

Proof of Lemma 9.20.1. By minimality, the graph $G' = G - \{uv\}$ admits an AVD $(k + 1)$ -coloring γ . If we create a K_2 connected component, then we can apply Lemma 9.19 on G . Otherwise, this coloring γ is a uv -partial safe edge-coloring of G , and Lemma 9.31 applies. By coloring uv with one of the compatible colors, we obtain an AVD $(k + 1)$ -coloring of G . \square

Proof of Lemma 9.20.2. Note that for all i , $d(v_i) \leq \lfloor \frac{k-1}{2} \rfloor$. By minimality, the graph $G' = G - \{v_1v_2, v_2v_3\}$ admits an AVD $(k + 1)$ -coloring γ . If one or more connected component is isomorphic to K_2 , then either v_2 is a cut-vertex and Lemma 9.19 applies, or v_2 is adjacent to a vertex of degree 1 and Lemma 9.20.1 applies. Otherwise, this coloring γ is a v_1v_2 -partial safe edge-coloring for G . Lemma 9.31 ensures that there is a v_1v_2 -partial coloring α of $G' = G - \{v_2v_3\}$ with $n_{compa}(\alpha) \geq 2$ as v_1 and v_2 have different degrees in G' which are both at most $\lfloor \frac{k-1}{2} \rfloor$. In other words there exist two AVD $(k + 1)$ -colorings α_1 and α_2 for G' , matching on all edges but v_1v_2 , and such that $\alpha_1(v_1v_2) \neq \alpha_2(v_1v_2)$. In particular, $\mathcal{C}_{v_3}(\alpha_1) = \mathcal{C}_{v_3}(\alpha_2)$ and $\mathcal{C}_{v_2}(\alpha_1) \neq \mathcal{C}_{v_2}(\alpha_2)$. Thus we can assume w.l.o.g. that $\mathcal{C}_{v_2}(\alpha_1) \neq \mathcal{C}_{v_3}(\alpha_1)$. Hence Lemma 9.31 ensures that α_1 can be extended to a safe edge-coloring to the whole graph G . \square

```

1 Uncolor  $uu_1$ ;
2 Let  $x$  be a vertex in conflict with  $u_1$  for color  $c_x$  such that  $c_x \in \{c_1, c_2\}$ ;
3 while  $uu_1$  cannot be colored with  $c_1$  or  $c_2$  do
4   | Let  $y$  be a vertex in conflict with  $u_1$  for color  $c_y$  such that  $c_y \in \{c_1, c_2\}$  and
   |    $c_x \neq c_y$ ;
5   | Recolor  $u_1y$  with a color distinct from  $c_1, c_2$ , and from the current color of  $u_1y$ 
   |   (and distinct from  $\gamma(uu_1)$  when  $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$ );
6   | Set  $x$  to be the current vertex  $y$ ;
7 end
8 Recolor  $uu_1$  with  $c_1$  or  $c_2$ ;

```

Algorithm 2: Recoloring uu_1 .

The rest of this section is devoted to the proof of Lemma 9.31. We start by presenting a recoloring algorithm in Lemma 9.32 which can be used when the number of common colors ρ is “small”. After using the previous algorithm on our coloring, we present a simpler algorithm but quite technical in its formulation in Claim 9.38 when the number of common colors is “large”. We conclude by proving Lemma 9.31 by using the previous recoloring algorithms and dividing the proof into a number of subcases.

9.4.3 Recoloring algorithm

In this subsection, we present a recoloring algorithm that will help us in the rest of the proof.

Lemma 9.32. *Let G be a graph and u, v be two vertices such that $d(u) \leq \lfloor \frac{k}{2} \rfloor$. Let γ be a uv -partial safe edge-coloring, such that u has a neighbor u_1 that is in a γ -conflict with u when uv is colored a .*

If $c_1, c_2 \notin \mathcal{C}_u(\gamma) \cup \{a\}$ are two colors, then there exists a uv -partial almost-safe edge-coloring α such that all edges e non-incident with u_1 verify $\alpha(e) = \gamma(e)$, $\alpha(u_1v) = \gamma(u_1v)$ (if the edge exists) and such that $\alpha(uu_1) \in \{c_1, c_2\}$.

Moreover when $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$, if u_1z was recolored, then we can suppose that $\alpha(u_1z) \neq \gamma(uu_1)$ and $\gamma(uu_1) \notin \mathcal{C}_z(\gamma)$.

This lemma is used to increase the number of colors which appear on both u and v . With the right choice of colors c_1 and c_2 , e.g. $c_1, c_2 \in \mathcal{C}_v(\gamma) \setminus \mathcal{C}_u(\gamma)$, we can add one more common color between the two vertices. Note that we do not require that $\mathcal{C}_u(\beta) \neq \mathcal{C}_v(\beta)$ (β is almost-safe and not necessarily safe), indeed this property will be a consequence of the choice of c_1 and c_2 .

Proof. Since u and u_1 are in a γ -conflict when uv is colored a , Observation 9.15 ensures that $\mathcal{C}_{u_1} \setminus \mathcal{C}_u = \{a\}$ and the definition ensures that $a \notin \mathcal{C}_v(\gamma)$. Let $c_1, c_2 \notin \mathcal{C}_u(\gamma) \cup \{a\}$. The coloring α is obtained from γ by recoloring only edges incident with u_1 (except u_1v if it exists), as described in Algorithm 2.

We first prove that instructions of Algorithm 2 are well-defined, and then that Algorithm 2 ends. A *step* will denote the sequence of operations made during an iteration of the while loop. The vertices x and y are said to be *relative to the step*.

Claim 9.33. *Algorithm 2 is well-defined.*

Proof. At the beginning, uu_1 cannot be recolored with c_1 or c_2 . Remember that $d(u_1) = d(u)$, and $\mathcal{C}_{u_1}(\gamma) \setminus \mathcal{C}_u(\gamma) = \{a\}$, with $a \notin \mathcal{C}_v(\gamma)$. In particular, $c_1, c_2 \notin \mathcal{C}_{u_1}(\gamma)$. Therefore, a neighbor of u_1 , denoted by x , in conflict with u_1 for the color $c_x \in \{c_1, c_2\}$, exists.

At each iteration of the while loop, uu_1 cannot be recolored with c_1 or c_2 and u_1 is in conflict with x . Since we only recolor edges incident with u_1 , with colors different from c_1 and c_2 , both c_1 and c_2 are not part of \mathcal{C}_{u_1} at this step of the algorithm. Therefore, a neighbor of u_1 , denoted by y , in conflict with u_1 for the color $c_y \in \{c_1, c_2\}$ with $c_y \neq c_x$, exists. Note that by definition of a conflict in that case, this means that all the edges incident with y are colored, and in particular that $y \neq v$ (the same holds for x). Note that $\mathcal{C}_x = \mathcal{C}_{u_1} \cup \{c_x\}$ and $\mathcal{C}_y = \mathcal{C}_{u_1} \cup \{c_y\}$ since uu_1 is uncolored. Hence both x and y are well-defined and are distinct since they are not incident with the same set of colors. Uncolor u_1y .

By Observation 9.15, $d(y) = d(u_1) \leq \lfloor \frac{k}{2} \rfloor$. Hence at most $d(y) - 1$ colors are incident with u_1y . There remains a set A of $k + 2 - d(y)$ colors non-incident with u_1y .

At most $d(y) - 1$ neighbors may be in conflict with y or u_1 for some color of xu_1 . Indeed, no neighbor of u_1 can be in conflict with u_1 for some color of yu_1 since uu_1 is uncolored and y has $d(y) - 1$ neighbors distinct from u_1 . Therefore, we can set A' to be the colors of A that are compatible with yu_1 and we have $|A'| \geq k + 3 - 2d(y)$. Additionally, $c_y \notin A'$ since there must be an edge incident with y (other than u_1y) that is colored in c_y . We remove c_x from A' and the previous color of u_1y from A' (if they are included in this set). Now there remains at least one color compatible with u_1y in A' since $k + 3 - 2d(x) - 2 \geq 1$ as $d(y) \leq \lfloor \frac{k}{2} \rfloor$. Moreover when $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$, $k + 3 - 2d(x) - 2 \geq 2$, thus we can forbid one extra color, namely $\gamma(uu_1)$ on the edge uy . ■

At the end of a step, by construction, the new color of u_1y does not belong to \mathcal{C}_x . Indeed, x was in conflict with u_1 for a coloring of u_1u by c_x , thus before the recoloring, $\mathcal{C}_x = \mathcal{C}_{u_1} \cup \{c_x\}$. Then we recolored u_1y with a color different from c_x , and from any element of \mathcal{C}_{u_1} before the recoloring, thus different from any element of \mathcal{C}_x (both before and after the recoloring, since the coloring is proper). Hence x cannot be in conflict with u_1 for any color of uu_1 as both the new color of u_1y and c_x do not belong to \mathcal{C}_{u_1} .

Claim 9.34. *Algorithm 2 ends.*

Proof. Let us prove by contradiction a stronger statement: no edge u_1w is recolored twice during the algorithm. Consider by contradiction the first vertex w such that u_1w is recolored twice. Let i and j be the rank of the corresponding steps where the edge is recolored with $i < j$. In the step $i + 1$, w plays the role of x and there is some vertex r which plays the role of y . Let us denote by γ_i the coloring at the end of first step i , γ_{i+1} the coloring at the end of the step $i + 1$, and γ_{j-1} the coloring at the beginning of the step j . When u_1w is recolored a second time, at step j , it must play the role of y thus step j must occur after step $i + 1$.

Since edges recolored during Algorithm 2 are edges incident with u_1 and since w is the first vertex which is recolored at least twice during the algorithm, we have $\gamma_{i+1}(u_1r) = \gamma_{j-1}(u_1r)$, and $\mathcal{C}_w(\gamma_i) = \mathcal{C}_w(\gamma_{i+1}) = \mathcal{C}_w(\gamma_{j-1})$. However, as argued before Claim 9.34, we have $\gamma_{i+1}(u_1r) \notin \mathcal{C}_w(\gamma_{i+1})$. Thus w cannot be in conflict with u_1 for any choice of color for uu_1 at step j , a contradiction.

So every edge is recolored at most once. Therefore the condition of the while is satisfied in at most $d(u) - 1$ steps. Therefore, Algorithm 2 ends. ■

At the end of the algorithm, we obtain a uv -partial safe edge-coloring α . Only edges incident with u_1 and not with v were recolored in the algorithm. Moreover, uu_1 is now colored with color c_1 or c_2 . Suppose that $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$. If u_1z was recolored, then $\alpha(u_1z) \neq \gamma(uu_1)$ as we forbid this color. Moreover if $\gamma(uu_1) \in \mathcal{C}_z(\gamma)$, then z cannot be in conflict with u_1 as, for any coloring of uu_1 among c_1 and c_2 , $\gamma(uu_1) \notin \mathcal{C}_{u_1}$. \square

9.4.4 Proof of Lemma 9.31

We start this section by providing some useful observations that we will use in our computations (x is any real number):

- $2d(v) \leq k - 1 + \delta$,
- $x - 1 \leq \lfloor x \rfloor \leq x$, $x \leq \lceil x \rceil \leq x + 1$ and $\lfloor x + 1 \rfloor = \lfloor x \rfloor + 1$,
- $x = \lfloor x \rfloor + \lceil x \rceil$,
- $2 \lfloor \frac{x}{2} \rfloor \geq x - 1$ and $2 \lceil \frac{x}{2} \rceil \leq x + 1$.

Proof of Lemma 9.31. Let γ be a uv -partial safe edge-coloring maximizing $\varrho = |C^{com}(\gamma)|$. Its existence is given by the hypothesis of Lemma 9.31.

We start the proof by proving a useful claim.

Claim 9.35. *Suppose that γ is a uv -partial safe edge-coloring maximizing $\varrho = |C^{com}(\gamma)|$. If $d(v) \geq 3 + \varrho$ (resp. $d(u) \geq 3 + \varrho$) and x is in a γ -conflict with u (resp. v), then $\gamma(ux) \in \mathcal{C}^{com}(\gamma)$ (resp. $\gamma(vx) \in \mathcal{C}^{com}(\gamma)$).*

Proof of Claim 9.35. Suppose it is not the case. Take any such vertex $x \in S_u$, i.e. x is in a γ -conflict with u for some color $a \notin \mathcal{C}_u(\gamma) \cup \mathcal{C}_v(\gamma)$. As $d(v) \geq 3 + \varrho$, v has at least two neighbors y_1, y_2 different from u such that $\gamma(vy_1), \gamma(vy_2) \notin \mathcal{C}_u(\gamma)$.

By Lemma 9.32 applied to u , $u_1 = x$, $c_1 = \gamma(vy_1)$ and $c_2 = \gamma(vy_2)$ (which is possible since $a \neq c_1$ and $a \neq c_2$), there is a uv -partial almost-safe edge-coloring α such that $|C^{com}(\alpha)| = \varrho(\gamma) + 1$ as we replace an edge which is not in $C^{com}(\gamma)$ by c_1 or c_2 and no other edge adjacent to u or v was recolored. Moreover $\mathcal{C}_u(\alpha) \neq \mathcal{C}_v(\alpha)$ as one of c_1 or c_2 (the one which was not assigned to ux) belongs to $\mathcal{C}_v(\alpha) \setminus \mathcal{C}_u(\alpha)$. Thus α is a uv -partial safe edge-coloring with $|C^{com}(\alpha)| \geq \varrho + 1$, a contradiction with the definition of γ . \blacksquare

Our following recoloring claim shows the existence of a coloring maximizing ϱ with a small number of conflict for vertex v under some favorable conditions.

Claim 9.36. *Suppose that γ is a uv -partial safe edge-coloring maximizing $\varrho = |C^{com}(\gamma)|$. If ϱ is at most $d(u) - 3$, then there exists a uv -partial safe edge-coloring α also maximizing ϱ such that $F_v(\alpha) \leq \lfloor \frac{\varrho+1}{2} \rfloor$.*

Proof of Claim 9.36. Suppose it is not the case, then $|S_v(\gamma)| \geq \lfloor \frac{\varrho+1}{2} \rfloor + 1$. Take any vertex $x \in S_v(\gamma)$ such that $\gamma(xv) \in \mathcal{C}^{com}(\gamma)$, i.e. x is in a γ -conflict with v for some color $a \notin \mathcal{C}_u(\gamma) \cup \mathcal{C}_v(\gamma)$. It is possible to make such a choice by Claim 9.35. As $d(u) \geq 3 + \varrho$, the vertex u has at least two neighbors y_1, y_2 different from v such that $\gamma(uy_1), \gamma(uy_2) \notin \mathcal{C}_v(\gamma)$.

By Lemma 9.32 applied to $u = v$, $u_1 = x$, $c_1 = \gamma(uy_1)$ and $c_2 = \gamma(uy_2)$ (which is possible since $a \neq c_1$ and $a \neq c_2$), there is a uv -partial almost-safe edge-coloring α with the properties of Lemma 9.32. Note that $\mathcal{C}_u(\alpha) \neq \mathcal{C}_v(\alpha)$, indeed one of c_1 or c_2 (the one which was not assigned to vx) belongs to $\mathcal{C}_u(\alpha) \setminus \mathcal{C}_v(\alpha)$. Thus α is a uv -partial safe

edge-coloring. Moreover α maximizes ϱ as we remove one common color $\gamma(vx)$ and we add one new common color $\alpha(vx)$.

By Claim 9.35, if v is in conflict with z , then $\alpha(vz) \in \mathcal{C}^{com}(\alpha)$. The vertex v might be in a α -conflict with x but not with any vertex of $S_v(\gamma) \setminus \{x\}$. Indeed if $z \in S_v(\gamma) \setminus \{x\}$ is in a α -conflict with v , then $\mathcal{C}_z(\alpha) = \mathcal{C}_v(\alpha) \cup \{b\}$ for some colors b . Remark that $\gamma(vx) \in \mathcal{C}_z(\gamma)$ (since $z \in S_v(\gamma)$), zx (if it exists) is the only edge incident with z which might have changed color between the two colorings and in this case $\gamma(zx) \neq \gamma(vx)$ (since the edges are adjacent). Therefore $\gamma(vx) \in \mathcal{C}_z(\alpha)$. This implies that $b = \gamma(vx)$ as $\gamma(vx) \notin \mathcal{C}_v(\alpha)$. In particular $b \in \mathcal{C}_u(\alpha)$ by the choice of x , a contradiction with the definition of a α -conflict.

Finally, as $2 \lfloor \frac{x}{2} \rfloor \geq x - 1$:

$$F_v(\alpha) \leq \varrho - (|S_v(\gamma)| - 1) \leq \varrho - \left(\left\lfloor \frac{\varrho + 1}{2} \right\rfloor + 1 - 1 \right) \leq \varrho - \left\lfloor \frac{\varrho + 1}{2} \right\rfloor \leq \left\lfloor \frac{\varrho + 1}{2} \right\rfloor.$$

■

Now, we show that we can do the same for vertex u . We must be careful to not introduce more conflicts for vertex v .

Claim 9.37. *Suppose that γ is a uv -partial safe edge-coloring maximizing $\varrho = |\mathcal{C}^{com}(\gamma)|$. If ϱ is at most $d(v) - 3$, then there exists a uv -partial safe edge-coloring α also maximizing ϱ such that $F_u \leq \lfloor \frac{\varrho + 1}{2} \rfloor$ and such that F_v is non-increasing.*

Proof. This proof is sensibly the same as the proof of Claim 9.36. Therefore, we just need to prove that F_v is non-increasing in the process. For this, we dispose of the extra assumption $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$.

Suppose the result does not hold when $\alpha = \gamma$, hence $|S_u(\gamma)| \geq \lfloor \frac{\varrho + 1}{2} \rfloor + 1$. Take any vertex $x \in S_u(\gamma)$ such that $\gamma(uv) \in \mathcal{C}^{com}(\gamma)$. As $d(v) \geq 3 + \varrho$, v has at least two neighbors y_1, y_2 different from u such that $\gamma(vy_1), \gamma(vy_2) \notin \mathcal{C}_u(\gamma)$.

By Lemma 9.32 applied to u , $u_1 = x$, $c_1 = \gamma(vy_1)$ and $c_2 = \gamma(vy_2)$, there is a uv -partial almost-safe edge-coloring α with the properties (and extra properties) of the lemma since $d(u) \leq \lfloor \frac{k-1}{2} \rfloor$. As before, α is a safe edge-coloring and we can show that $F_u \leq \lfloor \frac{\varrho + 1}{2} \rfloor$.

It is left to show that no new vertex z becomes in conflict with v after the recoloring. Note first that x is not incident with color $\gamma(ux)$ in α and $\gamma(ux) \in \mathcal{C}_v(\gamma) = \mathcal{C}_v(\alpha)$, hence x cannot be in conflict with v .

Suppose there exists $z \neq x$ in a α -conflict with v but not in a γ -conflict with v . The set of non-incident colors with uv is unchanged since $\gamma(ux) \in \mathcal{C}_v(\gamma) = \mathcal{C}_v(\alpha)$ and $c_1, c_2 \in \mathcal{C}_v(\gamma)$, thus \mathcal{C}_z changed and xz was recolored. As $\gamma(ux) \in \mathcal{C}_v(\gamma) = \mathcal{C}_v(\alpha)$, $\gamma(ux) \in \mathcal{C}_z(\alpha)$. By Lemma 9.32, xz was not recolored by $\gamma(ux)$ thus $\gamma(ux) \in \mathcal{C}_z(\gamma)$. By Lemma 9.32, xz was not recolored, a contradiction. ■

We have the necessary tools to resolve our first case.

Proof of case: $\varrho \leq d(u) - 3$. As $d(u) \leq d(v)$, remark that $\varrho \leq d(v) - 3$. We start by applying Claim 9.36 to γ to get a coloring α . We then apply Claim 9.37 to α to get a coloring β . The coloring β is a uv -partial safe edge-coloring by these two claims. Moreover, β maximizes $|\mathcal{C}^{com}| \leq d(u) - 3$.

The number of incident colors with uv in β is $n_i(\beta) = (d(u) - 1) + (d(v) - 1) - \varrho$. The number of β -conflicts for uv is at most $F_u(\beta) + F_v(\beta)$. By Claim 9.36 and Claim 9.37, both $F_u(\beta)$ and $F_v(\beta)$ are at most $\lfloor \frac{\varrho + 1}{2} \rfloor$.

As $d(u) \leq d(v) \leq \lfloor \frac{k}{2} \rfloor$, the number of compatible colors with uv is at least:

$$\begin{aligned}
 n_{\text{compa}}(\beta) &\geq k + 1 - n_i(\beta) - (F_u(\beta) + F_v(\beta)) \\
 &\geq k + 3 - d(u) - d(v) + \varrho - 2 \left\lfloor \frac{\varrho + 1}{2} \right\rfloor \\
 &\geq k + 3 - \left\lfloor \frac{k}{2} \right\rfloor - \left\lfloor \frac{k}{2} \right\rfloor + \varrho - (\varrho + 1) \\
 &\geq k + 2 - k \\
 &\geq 2.
 \end{aligned}$$

■

Claim 9.38. *Suppose that γ is a uv -partial safe edge-coloring. Suppose $\varrho = d(u) - 1$ or $d(u) - 2$. Let $x \in S_v(\gamma)$ and c_f be an arbitrary color different from $\gamma(vx)$. There exists a uv -partial almost-safe edge-coloring α such that either, $\alpha = \gamma$ and $F_v(\gamma) \leq \lceil \frac{d(v)}{2} \rceil$, or all the following properties hold.*

1. $|Conf_v(\alpha) \setminus \{\gamma(vx)\}| \leq \lceil \frac{d(v)}{2} \rceil - 1$.
2. γ and α differ only on vx .
3. $\alpha(vx) \neq c_f$.
4. If $\gamma(vx) \in \mathcal{C}^{\text{com}}(\gamma)$, then
 - (a) $|\mathcal{C}^{\text{com}}(\alpha)| = |\mathcal{C}^{\text{com}}(\gamma)| - i$ where $i \in \{0, 1\}$,
 - (b) $S_u(\alpha) \subseteq S_u(\gamma)$, and
 - (c) $\gamma(vx) \notin \mathcal{C}^{\text{com}}(\alpha)$.
5. If $\gamma(vx) \notin \mathcal{C}^{\text{com}}(\gamma)$, then
 - (a) $|\mathcal{C}^{\text{com}}(\alpha)| \geq |\mathcal{C}^{\text{com}}(\gamma)|$,
 - (b) $S_u(\alpha) \subseteq S_u(\gamma) \cup \{x\} \cup A$ where A is the subset of neighbors of u in conflict with u when uv is colored $\gamma(vx)$.

The same holds by exchanging u with v .

Proof of Claim 9.38. Suppose $|S_v(\gamma)| \geq F_v(\gamma) \geq \lceil \frac{d(v)}{2} \rceil + 1$. We note $a = \gamma(vx)$.

Uncolor vx . There are at least $k + 1 - 2(d(x) - 1) \geq 3$ colors compatible with this edge, as $\mathcal{C}_v(\gamma) \subseteq \mathcal{C}_x(\gamma)$ and x cannot be in conflict with v since it has an uncolored edge. Choose a color for vx which is not a nor c_f . Let α be the obtained uv -partial almost-safe edge-coloring.

As all vertices of $S_v \setminus \{x\}$ are adjacent to a (as they were in conflict before), they can only be in conflict with v if uv is colored a . Hence all of them can only remove one possible color for uv which is a . Therefore:

$$|Conf_v(\alpha) \setminus \{a\}| \leq d(v) - 1 - (|S_v| - 1) \leq d(v) - \left\lceil \frac{d(v)}{2} \right\rceil - 1 \leq \left\lfloor \frac{d(v)}{2} \right\rfloor - 1.$$

This concludes the proof of Claim 9.38.1-3.

Suppose $a \in C_u(\gamma)$. We remove one common color between u and v , depending on whether $\alpha(vx) \in C^{com}(\alpha)$ or not, we get the two possibilities for Claim 9.38.4a. Claim 9.38.4c follows from the fact that we choose $\alpha(vx)$ different from $\gamma(vx)$ and all the other colors of edges incident with v . Suppose that $z \in S_u(\alpha) \setminus S_u(\gamma)$. As only vx changed and all non-incident colors with uv in γ are non-incident in α (as an edge of u is colored a), we have $z = x$. The edge ux is not colored a since it is adjacent to vx and $\alpha(vx) \neq a$. As x and u are in conflict, $a \in C_x(\alpha)$ which implies that x has two incident edges colored a in γ , a contradiction. This proves Claim 9.38.4b.

Suppose $a \notin C_u(\gamma)$. We remove no common color between u and v , depending on whether $\alpha(vx) \in C^{com}(\alpha)$ or not, we might add one, hence we get Claim 9.38.5a. Suppose that $z \in S_u(\alpha) \setminus (S_u(\gamma) \cup \{x\})$. As only vx changed and the only non-incident color with uv in α which is incident in γ is a , we obtain $C_z(\alpha) = C_u(\alpha) \cup \{a\}$. Indeed recall that z had none of its incident recolored edges. This proves Claim 9.38.5b. ■

The cases where S_v or S_u are empty sets are simple cases where we take $\alpha = \gamma$ in the previous claims. Finally, we can prove our result in the last three cases.

Proof of case: $\varrho = d(u) - 2$ and $d(u) < d(v)$. In this case, $\varrho \leq d(v) - 3$ as $d(u) < d(v)$.

We apply Claim 9.37 to γ to get a coloring α . The number of incident colors with uv in α is $n_i(\alpha) = d(u) + d(v) - \varrho - 2 = d(v)$ and $F_u(\alpha) \leq \lfloor \frac{d(u)-1}{2} \rfloor$. The coloring α is a uv -partial safe edge-coloring.

Take an arbitrary vertex $x \in S_v$, we apply Claim 9.38 to α to get a coloring β . As $d(u) \neq d(v)$, β is a uv -partial safe edge-coloring. If $\alpha = \beta$, then $F_v(\beta) \leq \lfloor \frac{d(v)}{2} \rfloor$. When $\alpha \neq \beta$, we have two more cases depending on whether $\alpha(vx) \in C_u(\alpha)$ or not. If $\alpha(vx) \in C_u(\alpha)$, then by Claim 9.38, $F_v(\beta) \leq \lfloor \frac{d(v)}{2} \rfloor$ (1), $F_u(\beta) \leq F_u(\alpha) \leq \lfloor \frac{d(u)-1}{2} \rfloor$ (4b) and $|C^{com}(\beta)| \geq |C^{com}(\alpha)| - 1$ (4a). This last point implies $n_i(\beta) \leq n_i(\alpha) + 1$. If $\alpha(vx) \notin C_u(\alpha)$, then by Claim 9.38, $F_v(\beta) \leq \lfloor \frac{d(v)}{2} \rfloor$ (1), $F_u(\beta) \leq \lfloor \frac{d(u)-1}{2} \rfloor + 1$ (5b) as the set A is a set of vertices in conflict with u for the same color of uv . Moreover $n_i(\beta) \leq n_i(\alpha) \leq d(v)$ (5a).

Hence, in all cases, and as $d(v) \leq \lfloor \frac{k}{2} \rfloor$, the number of compatible colors with uv in β is at least:

$$\begin{aligned} n_{compa}(\beta) &\geq k + 1 - n_i(\beta) - F_u(\beta) - F_v(\beta) \\ &\geq k + 1 - d(v) - \left\lfloor \frac{d(u) - 1}{2} \right\rfloor - \left\lfloor \frac{d(v)}{2} \right\rfloor - 1 \\ &\geq k - d(v) - \left\lfloor \frac{d(v)}{2} \right\rfloor + 1 - \left\lfloor \frac{d(v)}{2} \right\rfloor \\ &\geq k + 1 - 2d(v) \\ &\geq k + 1 - (k - 1 + \delta) \\ &\geq 2 - \delta. \end{aligned}$$

■

Proof of case: $\varrho = d(u) - 2$ and $d(u) = d(v)$. In this case, $d(v) = d(u) \leq \lfloor \frac{k-1}{2} \rfloor$. The number of incident colors with uv is $n_i(\gamma) = d(u) = d(v)$.

Suppose there is no color a such that, for some vertices x and y , u is in a γ -conflict with x where $\gamma(ux) = a$ and v is in a γ -conflict with y where $\gamma(vy) = a$. Then, as each

color in $\mathcal{C}^{com}(\gamma)$ cannot be both on an edge ux and vy for $x \in S_u(\gamma)$ and $y \in S_v(\gamma)$: $F_u + F_v \leq (d(u) - 1 - \varrho) + (d(v) - 1 - \varrho) + \varrho = d(u)$. Thus the number of compatible colors is at least:

$$\begin{aligned} n_{compa}(\gamma) &\geq k + 1 - n_i(\gamma) - F_u(\gamma) - F_v(\gamma) \\ &\geq k + 1 - d(u) - d(u) \\ &\geq 2. \end{aligned}$$

Suppose now that such a color exists. Let x and y such that u is in a γ -conflict with x where $\gamma(ux) = a$ and v is in a γ -conflict with y where $\gamma(vy) = a$.

Choose $b_u \in \mathcal{C}_u(\gamma) \setminus \mathcal{C}_v(\gamma)$ (which exists since γ is safe) and remark that $a \neq b_u$. We apply Claim 9.38 to γ , v and y to get a coloring α with $c_f = b_u$.

- **Suppose $\alpha = \gamma$.** Choose $b_v \in \mathcal{C}_v(\alpha) \setminus \mathcal{C}_u(\alpha)$ (which exists since $\alpha = \gamma$ is safe) and remark that $a \neq b_v$. We apply Claim 9.38 to α , u and x to get a coloring β where $c_f = b_v$. Let $\varepsilon = 1$ when $\beta = \alpha$ and 0 otherwise. We use ε to remove the case analysis (the arguments will be given for the case $\alpha \neq \beta$). By Claim 9.38, properties (2) and (3), $b_v \in \mathcal{C}_v(\beta) \setminus \mathcal{C}_u(\beta)$. In particular, $\mathcal{C}_u(\beta) \neq \mathcal{C}_v(\beta)$, hence, β is safe. We have $|Conf_u(\beta)| \leq \left\lceil \frac{d(u)}{2} \right\rceil - 1 + \varepsilon$ (1) as $a \notin Conf_u(\beta)$ since $a \in \mathcal{C}_v(\beta)$. Moreover, $n_i(\beta) \leq n_i(\gamma) + 1 - \varepsilon$ (4a) and $F_v(\beta) \leq F_v(\gamma) \leq \left\lceil \frac{d(v)}{2} \right\rceil$ (4b). Hence,

$$\begin{aligned} n_{compa}(\beta) &\geq k + 1 - n_i(\beta) - F_u(\beta) - F_v(\beta) \\ &\geq k + 1 - d(v) - 1 + \varepsilon - 2 \left\lceil \frac{d(v)}{2} \right\rceil + 1 - \varepsilon \\ &\geq k + 1 - 2d(v) - 1 \\ &\geq 1. \end{aligned}$$

- **Suppose $\alpha \neq \gamma$.** By Claim 9.38, properties (2) and (3), $b_u \in \mathcal{C}_u(\alpha) \setminus \mathcal{C}_v(\alpha)$. In particular, $\mathcal{C}_u(\alpha) \neq \mathcal{C}_v(\alpha)$, hence, α is safe. We have, $|Conf_v(\alpha) \setminus \{a\}| \leq \left\lceil \frac{d(v)}{2} \right\rceil - 1$ (1), $n_i(\alpha) \leq n_i(\gamma) + 1$ (4a) and $F_u(\alpha) \leq F_u(\gamma)$ (4b).

Recall that $\alpha(ux) = \gamma(vy) \notin \mathcal{C}^{com}(\alpha)$ by (4c). Choose $b_v \in \mathcal{C}_v(\alpha) \setminus \mathcal{C}_u(\alpha)$ and remark that $a \neq b_v$. Therefore, we apply Claim 9.38 to α , u and x with $c_f = b_v$ to get a coloring β . If $\beta = \alpha$, then set $\varepsilon = 1$, and 0 otherwise. By Claim 9.38, properties (2) and (3), $b_v \in \mathcal{C}_v(\beta) \setminus \mathcal{C}_u(\beta)$. In particular, $\mathcal{C}_u(\beta) \neq \mathcal{C}_v(\beta)$ and β is safe. Moreover, $|Conf_u(\beta) \setminus \{a\}| \leq \left\lceil \frac{d(u)}{2} \right\rceil - 1 + \varepsilon$ (1), $n_i(\beta) \leq n_i(\alpha)$ (5a) and $|Conf_v(\beta) \setminus \{a\}| \leq \left\lceil \frac{d(v)}{2} \right\rceil - 1$ (5b) as x cannot be in conflict with v since $b_v \in \mathcal{C}_v(\beta)$ and $b_v \notin \mathcal{C}_x(\beta)$. If $\beta = \alpha$, then $a \in \mathcal{C}_u(\beta)$ and thus a is not in $Conf_u(\beta)$ nor $Conf_v(\beta)$.

As $d(v) \leq \left\lceil \frac{k-1}{2} \right\rceil$, the number of compatible colors with uv is at least (the last term

of the first line is an upper bound on whether a belongs to $Conf_u(\beta) \cup Conf_v(\beta)$:

$$\begin{aligned}
 n_{compa}(\beta) &\geq k + 1 - n_i(\beta) - |Conf_u(\beta) \setminus \{a\}| - |Conf_v(\beta) \setminus \{a\}| - (1 - \varepsilon) \\
 &\geq k + 1 - (d(v) + 1) - \left\lceil \frac{d(u)}{2} \right\rceil + 1 - \varepsilon - \left\lceil \frac{d(v)}{2} \right\rceil + 1 - 1 + \varepsilon \\
 &\geq k + 1 - d(v) - \left(\left\lceil \frac{d(v)}{2} \right\rceil + 1 \right) - \left\lceil \frac{d(v)}{2} \right\rceil \\
 &\geq k - 2d(v) \\
 &\geq k - (k - 1) \\
 &\geq 1.
 \end{aligned}$$

This is the desired number of possibilities when $d(u) = d(v)$. ■

Proof of case: $\varrho = d(u) - 1$. In this case, $d(u) \leq d(v) - 1$. In particular, all colorings are safe. The number of incident colors with uv is $n_i(\gamma) = d(v) - 1$.

Suppose there is no color a such that, for some vertices x and y , u is in a γ -conflict with x where $\gamma(ux) = a$ and v is in a γ -conflict with y where $\gamma(vy) = a$. Then, as in the previous case, $F_u + F_v \leq (d(u) - 1 - \varrho) + (d(v) - 1 - \varrho) + \varrho \leq d(v) - 1$. Thus the number of compatible colors is at least:

$$\begin{aligned}
 n_{compa}(\gamma) &\geq k + 1 - n_i(\gamma) - F_u - F_v \\
 &\geq k + 1 - d(v) + 1 - d(v) + 1 \\
 &\geq 3.
 \end{aligned}$$

Suppose now that such a color exists. Let x and y such that u is in a γ -conflict with x where $\gamma(ux) = a$ and v is in a γ -conflict with y where $\gamma(vy) = a$.

We apply Claim 9.38 to γ , v and y to get a coloring α .

- **Suppose $\alpha = \gamma$.** We apply Claim 9.38 to α , u and x to get a coloring β . Let $\varepsilon = 1$ if $\beta = \alpha$ and 0 otherwise. We use ε to remove the case analysis (the arguments will be given for the case $\alpha \neq \beta$). We have $|Conf_u(\beta)| \leq \left\lceil \frac{d(u)}{2} \right\rceil - 1 + \varepsilon$ (1) as $a \notin Conf_u(\beta)$ since $a \in C_v(\beta)$. Moreover, $n_i(\beta) \leq n_i(\gamma) + 1 - \varepsilon$ (4a) and $F_v(\beta) \leq F_v(\gamma) \leq \left\lceil \frac{d(v)}{2} \right\rceil$ (4b). Hence,

$$\begin{aligned}
 n_{compa}(\beta) &\geq k + 1 - n_i(\beta) - F_u(\beta) - F_v(\beta) \\
 &\geq k + 1 - d(v) + \varepsilon - \left\lceil \frac{d(v) - 1}{2} \right\rceil + 1 - \varepsilon - \left\lceil \frac{d(v)}{2} \right\rceil \\
 &\geq k + 2 - d(v) - \left\lceil \frac{d(v)}{2} \right\rceil - \left\lceil \frac{d(v)}{2} \right\rceil \\
 &\geq k + 2 - 2d(v) \\
 &\geq 2.
 \end{aligned}$$

- **Suppose $\alpha \neq \gamma$.** We have, $|Conf_v(\alpha) \setminus \{a\}| \leq \left\lceil \frac{d(v)}{2} \right\rceil - 1$ (1), $n_i(\alpha) \leq n_i(\gamma) + 1$ (4a) and $F_u(\alpha) \leq F_u(\gamma)$ (4b).

Recall that $\alpha(ux) = \gamma(vy) \notin C^{com}(\alpha)$ by (4c). Therefore, we apply Claim 9.38 to α , u and x to get a coloring β . If $\beta = \alpha$, then set $\varepsilon = 1$, and 0 otherwise. We have,

$|Conf_u(\beta) \setminus \{a\}| \leq \left\lceil \frac{d(u)}{2} \right\rceil - 1 + \varepsilon$ (1), $n_i(\beta) \leq n_i(\alpha)$ (5a) and $|Conf_v(\beta) \setminus \{a\}| \leq \left\lceil \frac{d(v)}{2} \right\rceil - 1$ (5b) as x and v cannot be in conflict since they do not have the same degree. If $\beta = \alpha$, then $a \in \mathcal{C}_u(\beta)$ and thus a is not in $Conf_u(\beta)$ nor $Conf_v(\beta)$.

Hence, the number of compatible colors with uv is at least (the last term of the first line is an upper bound on whether a belongs to $Conf_u(\beta) \cup Conf_v(\beta)$):

$$\begin{aligned}
 n_{compa}(\beta) &\geq k + 1 - n_i(\beta) - |Conf_u(\beta) \setminus \{a\}| - |Conf_v(\beta) \setminus \{a\}| - (1 - \varepsilon) \\
 &\geq k + 1 - d(v) - \left\lceil \frac{d(v) - 1}{2} \right\rceil + 1 - \varepsilon - \left\lceil \frac{d(v)}{2} \right\rceil + 1 - 1 + \varepsilon \\
 &\geq k + 2 - d(v) - \left\lfloor \frac{d(v)}{2} \right\rfloor - \left\lceil \frac{d(v)}{2} \right\rceil \\
 &\geq k + 2 - 2d(v) \\
 &\geq 2.
 \end{aligned}$$

■

This concludes the proof of Lemma 9.31. □

9.5 Discharging procedures

We can now present the proofs of our two results. Let us first present the ghost vertices method.

9.5.1 The ghost vertices method for mad discharging

The following observation will be useful for the proof of Lemma 9.17. Let $G = (V, E)$ be a graph. In the remaining of this section, m denote a real number greater than $\text{mad}(G)$. Recall that the weight function $w : V \rightarrow \mathbb{R}$ where $w(v) = d(v) - m$ has a global weight which is negative, *i.e.* $w(V) < 0$.

Observation 9.39. *Let w be the weight function where $w(v) = d(v) - m$. Let H be an induced subgraph of G . If there exists a weight function w' such that $w'(G) \leq w(G)$ and such that $w'(G - V(H)) \geq w(G - V(H)) + |E(H, G - V(H))|$, then $w'(H) < 0$.*

Proof. Since $m > \text{mad}(G)$, the function $w'' : H \rightarrow \mathbb{R}$ such that $w''(v) = d_H(v) - m$ is a function such that $w''(H) < 0$. The difference between $d(v)$ and $d_H(v)$ is exactly the number of neighbors of v in $G - V(H)$. So $\sum_{v \in H} d_H(v) = \sum_{v \in H} d(v) - |E(H, G - V(H))|$ and we have

$$w''(H) = \sum_{v \in H} (d_H(v) - m) = \sum_{v \in H} (d(v) - m) - |E(H, G - V(H))| = w(H) - |E(H, G - V(H))|.$$

By hypothesis $w'(G) \leq w(G)$. Using $w(G) = w(G - V(H)) + w(H)$, one can easily derive that $w'(H) \leq w(G - V(H)) + w(H) - w'(G - V(H))$. Using the second assumption of the observation, we obtain $w'(H) \leq w(H) - |E(H, G - V(H))| = w''(H)$. Hence we have $w'(H) \leq w''(H) < 0$. □



Figure 9.14: The discharging rules (R2) and (R3).

In other words, the vertices in $G - V(H)$ can be seen but, in a way, do not contribute to the sum analysis (the meaning of their final weight is essentially “this vertex has no positive contribution on the total weight of the rest of the graph”). This particularity leads us to informally refer to them as *ghost vertices*. Any result proved using ghost vertices can be proved, albeit more tediously perhaps, when deleting them completely from the graph. However, they can simplify the presentation of the discharging analysis, and this is the point of their introduction. See [20] for a reference on this technique.

9.5.2 Proof of Theorem 9.13

Proof of Lemma 9.17. We assign a weight of $w(u) = d(u) - m$ to every vertex u of G . Let H be the subgraph of G induced by the vertices of degree at least $\lfloor \frac{m}{2} \rfloor$. Each vertex u of $G - V(H)$ has at most $d(u)$ neighbors in H . Hence the number of edges in $E(H, G - V(H))$ is at most $\sum_{u \in G - V(H)} d(u)$. Hence a weight function w' where the weight of each vertex u of $G \setminus H$ is at least $2d(u) - m$ verifies $w'(G - V(H)) \geq w(G - V(H)) + |E(H, G - V(H))|$. We aim at redistributing the weights in the graph in such a way as to reveal a contradiction with $ad(H) < m$. By Observation 9.39, it suffices to ensure that every vertex u of H has a non-negative weight and each vertex of $G - V(H)$ has weight at least $2d(u) - m$.

For each integer i such that $\lfloor \frac{m}{2} \rfloor \leq i \leq \lceil m - 1 \rceil$, we create a “bank” $Bank_i$ with initial value 0. In the discharging procedure some vertices will give some of their weight to the banks and other vertices will receive some weight. To ensure that the total weight is non-increasing, we need to make sure that each bank has a non-negative weight after the discharging procedure.

We design the following discharging rules.

- (R1) Every vertex u of degree 1 receives 1 from its neighbor.
- (R2) For every i such that $\lfloor \frac{m}{2} \rfloor \leq i \leq \lceil m - 1 \rceil$, every vertex of degree at least $\lfloor \frac{k}{2} \rfloor + 1$ gives $4(i - 1)$ to $Bank_i$.
- (R3) For every i such that $\lfloor \frac{m}{2} \rfloor \leq i \leq \lceil m - 1 \rceil$, every vertex of degree at least 2 and at most i receives 1 from $Bank_i$.

We need to prove that the vertices of H receive a non-negative weight and vertices of $G - V(H)$ have a weight of at least $2d(u) - m$. Let w' be the new weight function after the discharging procedure. First note that $\frac{k}{2} > \frac{3}{2}(m + 1)^2 > m$. Remark that vertices of degree 1 can only have neighbors of degree at least $\lfloor \frac{k}{2} \rfloor + 1$ by Lemma 9.20. Thus, by the previous remarks, a vertex cannot both receive and give in the procedure.

Case $u \in G - V(H)$. If u is a vertex of $G - V(H)$, then it verifies $d(u) \leq \lfloor \frac{m}{2} \rfloor - 1 \leq \lfloor \frac{m}{2} \rfloor$.

- If $d(u) = 1$, then it receives 1 from its neighbor by (R1) thus $w'(u) = d(u) - m + 1 = 2d(u) - m$.

- If $d(u) \geq 2$, then u receives 1 from each bank. Thus, it verifies:

$$\begin{aligned} w'(u) &= w(u) + \sum_{i=\lceil m/2 \rceil}^{\lceil m-1 \rceil} 1 = d(u) - m + \lceil m-1 \rceil - \left\lfloor \frac{m}{2} \right\rfloor + 1 \\ &\geq d(u) - m + m - \left\lfloor \frac{m}{2} \right\rfloor \geq d(u) - m + \left\lfloor \frac{m}{2} \right\rfloor \geq 2d(u) - m. \end{aligned}$$

Case $u \in H$. In this case, $d(u) \geq \left\lfloor \frac{m}{2} \right\rfloor$.

- If $d(u) \leq \lceil m-1 \rceil$. As $d(u) \geq \left\lfloor \frac{m}{2} \right\rfloor$, u receives 1 from each bank of index at least $d(u)$. Thus:

$$\begin{aligned} w'(u) &= w(u) + \sum_{i=d(u)}^{\lceil m-1 \rceil} 1 = d(u) - m + \lceil m-1 \rceil - d(u) + 1 \\ &= \lceil m \rceil - m \geq 0. \end{aligned}$$

- If u is a vertex of H with degree at least $\lceil m-1 \rceil + 1 = \lceil m \rceil$ but less than $\left\lfloor \frac{k}{2} \right\rfloor + 1$, it receives nothing and gives nothing thus $w'(u) = w(u) = d(u) - m \geq \lceil m \rceil - m \geq 0$.
- If u is a vertex of H with degree at least $\left\lfloor \frac{k}{2} \right\rfloor + 1$. Note that $d(u) - d_1(u) \geq \left\lfloor \frac{k}{2} \right\rfloor + 1 \geq \frac{k}{2}$ by Lemma 9.23. Then, we obtain:

$$\begin{aligned} w'(u) &= d(u) - m - d_1(u) - \sum_{i=\lceil m/2 \rceil}^{\lceil m-1 \rceil} 4(i-1) \\ &\geq \frac{k}{2} - m - 4 \frac{(\lceil m-1 \rceil - 1 + \left\lfloor \frac{m}{2} \right\rfloor - 1) (\lceil m-1 \rceil - \left\lfloor \frac{m}{2} \right\rfloor + 1)}{2} \\ &\geq \frac{k}{2} - m - 2 \left(m + \frac{m}{2} + 1 - 2 \right) \left(m - \frac{m}{2} + 1 \right) \text{ since } \lceil x \rceil \leq x + 1 \\ &\geq \frac{k}{2} - m - 2 \left(\frac{3m}{2} - 1 \right) \left(\frac{m}{2} + 1 \right) \\ &\geq \frac{k}{2} - m - 2 \left(\frac{3m^2}{4} + m - 1 \right) \\ &\geq \frac{3}{2}(m+1)^2 - m - \frac{3m^2}{2} - 2m + 2 \text{ since } k > 3(m+1)^2 \\ &\geq \frac{7}{2}. \end{aligned}$$

Bank _{i} has non-negative weight. Now if i verifies $\left\lfloor \frac{m}{2} \right\rfloor \leq i \leq \lceil m-1 \rceil$, then $Bank_i$ receives $4(i-1)$ from each vertex in the set $B = \{u \mid d(u) \geq \left\lfloor \frac{k}{2} \right\rfloor + 1\}$ and gives 1 for each vertex in the set $D_i = \{u \mid 2 \leq d(u) \leq i\}$. In total $Bank_i = 4(i-1)|B| - |D_i|$ which is non-negative by Lemma 9.24 as $i \leq m < \left\lfloor \frac{k}{2} \right\rfloor$.

Thus $w'(G) \leq w(G)$ and Observation 9.39 implies $w'(H) < 0$, a contradiction. \square

Proof of Theorem 9.13. Let G be a connected graph with $\Delta(G) > 3 \cdot (\text{mad}(G) + 1)^2$. We can apply Lemma 9.17 to ensure that for every m such that $m > \text{mad}(G)$, G is AVD $(3(m + 1)^2)$ -colorable. By taking $m = \text{mad}(G) + \epsilon$, we obtain $3(m + 1)^2 = 3 \cdot (\text{mad}(G) + 1)^2 + 3\epsilon \cdot (2\text{mad}(G) + 2 + \epsilon) < \Delta(G)$, for a small enough ϵ , and hence our result. \square

Possible generalizations. Note that it is possible to obtain a $O(\text{mad}(G)^2)$ lower bound in the previous result with an easier discharging. Most notably, using a weaker version of Lemma 9.20 (*e.g.* where the $\frac{k}{2}$ bound is replaced by a $\frac{k}{4}$ bound) would yield a similar result with a worse multiplicative constant. In such a way, it would be possible to obtain a similar result for other types of vertex-distinguishing edge-colorings: total AVD coloring, neighbor sum distinguishing total coloring... The case of neighbor sum distinguishing coloring is more complicated as it is unclear how to obtain a result similar to Lemma 9.20.

9.5.3 Proof of Theorem 9.14

Let $k \geq 12$. We aim at proving Lemma 9.18, using a discharging method.

9.5.3.1 Discharging rules

Let G be a non P -reducible planar graph, we want to show that $G = G_\emptyset$.

Before moving to the actual discharging part of the proof, we make one observation that is a direct corollary of lemmas 9.20 and 9.25.1.

Observation 9.40. *In G , for every edge uv with $d(u) \leq 2$, we have $d(v) \geq 7$.*

In [22], Bonamy and Przybyło stated the planar version of Observation 9.39.

Observation 9.41 (reformulated from [22]). *Consider a planar graph G and a partition (V_1, V_2) of its vertices. For $i \in \{1, 2\}$, let G_i be the graph induced by V_i in G . Consider a mapping \mathcal{M} of G_1 to the plane. Assign to each vertex v of G a weight of $d_G(v) - 6$ and to each face of \mathcal{M} a weight of $2d(f) - 6$. It is not possible to discharge the weight over the graph in such a way that all the vertices and faces of \mathcal{M} have a non-negative weight, while every vertex v of V_2 has a weight of at least $d_G(v) + d_{G_1}(v) - 6$.*

We define G' as the graph G where all the vertices of degree 1 or 2 have been removed. Let \mathcal{M} be an embedding of G' on the plane.

In the following, the degree of a face is taken in \mathcal{M} , while the degree of a vertex is taken in G unless specified otherwise. Recall that a vertex v is a *weak* (resp. *semi-weak*) neighbor of a vertex u if both belong to G' and, in \mathcal{M} , the edge uv belongs to two (resp. exactly one) faces of degree 3. Recall that $c(v)$ is the number of vertices of degree $d(v)$ adjacent to v .

We assign to each vertex u of G a weight of $d_G(u) - 6$, and to each face f of \mathcal{M} a weight of $2d_{\mathcal{M}}(f) - 6$. We define the discharging rules as follows (see also Figure 9.15).

- (R_f) Every face f of degree at least 4 gives 1 to each 5^- -vertex incident with f .
- (R_1) Every 7^+ -vertex u gives 1 to every adjacent 2^- -vertex.
- (R_2) Every 7^+ -vertex u gives 1 to every adjacent weak 3-vertex v with $c(v) = 0$ and $\frac{1}{2}$ to every adjacent semi-weak 3-vertex v with $c(v) = 0$.

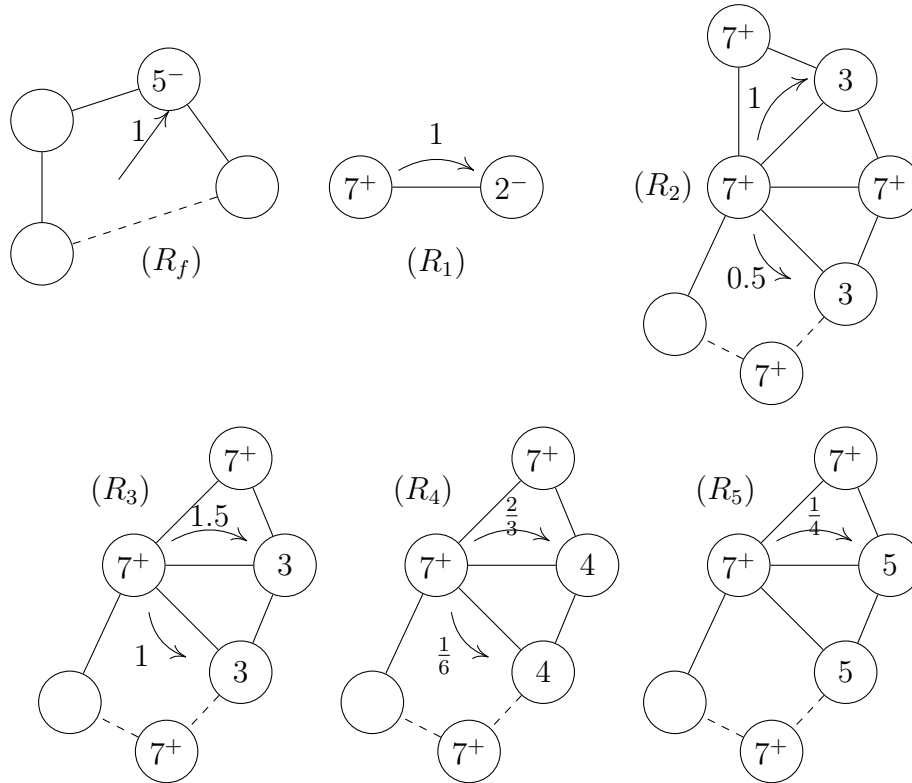


Figure 9.15: The discharging rules. Dashed edges represent arbitrary long path. The figures of (R_4) and (R_5) represent adjacent 4-vertices and adjacent 5-vertices in order to keep the figures small. The rules also apply when they are non-adjacent. Note that the adjacency or absence of adjacency between the 3-vertices is important for (R_2) and (R_3) . Hollow vertices may have additional edges and may coincide with other vertices.

- (R_3) Every 7^+ -vertex u gives $\frac{3}{2}$ to every adjacent weak 3-vertex v with $c(v) = 1$ and 1 to every adjacent semi-weak 3-vertex v with $c(v) = 1$.
- (R_4) Every 7^+ -vertex u gives $\frac{2}{3}$ to every adjacent weak 4-vertex and $\frac{1}{6}$ to every adjacent semi-weak 4-vertex.
- (R_5) Every 7^+ -vertex u gives $\frac{1}{4}$ to every adjacent weak 5-vertex.

Using Observation 9.41, in order to derive a contradiction with the existence of G , we aim at redistributing the weight along the graph as follows. Each vertex u of degree 1 or 2 has a weight of at least $2d(u) - 6$, while all the other vertices and all the faces have a non-negative weight. To that purpose, we apply all the rules, and prove that all the vertices and faces have the required weight.

First let f be a face of \mathcal{M} .

- If $d(f) = 3$, then f gives and receives no weight thus has final weight 0.
- If $d(f) = 4$, then f has at most two incident 5^- -vertices by Lemma 9.20 and has initial weight 2. The face f gives at most 2 by rule (R_f) and therefore has a non-negative final weight.
- If $d(f) \geq 5$, then f has at most $\lfloor \frac{2}{3}d(f) \rfloor$ incident 5^- -vertices by Lemma 9.20 and has initial weight $2d(f) - 6$. The face f gives at most $\lfloor \frac{2}{3}d(f) \rfloor$ by rule (R_f) and therefore has a final weight of at least $\frac{4}{3}d(f) - 6$ which is non-negative.

Now let u be a vertex of G .

- If $d(u) \leq 2$, then u has $d(u)$ neighbors, each of degree at least 7 by Observation 9.40. Thus u receives 1 from each neighbor by rule (R_1) and has final weight $2d(u) - 6$ which is the required weight.

All other vertices must have non-negative weight in order to use Observation 9.41. Let us start with 5^- -vertices which have initially a negative weight. Note that they do not have a 2^- -neighbor by Lemma 9.20. Moreover by Lemma 9.20, each 5^- -vertex has at most one adjacent 6^- -vertex. If $d(u) \leq 5$, then let a be the number of 4^+ -faces incident with u and t be the number of triangular faces incident with u . Let w be the number of weak neighbors of u and s be its number of semi-weak neighbors. Note that $a + t = d(u)$ and $2w + s = 2t$. Thus $d(u) = a + w + \frac{s}{2}$.

- If $d(u) = 3$ and $c(u) = 0$, then u must receive at least 3. By (R_f) and (R_2) , u receives exactly $a + w + \frac{s}{2}$ as u has only 7^+ -neighbors by Lemma 9.20 and the hypothesis $c(u) = 0$. This is equal to 3 in this case, so u has non-negative final weight.
- If $d(u) = 3$ and $c(u) = 1$, then u must receive at least 3. By Lemma 9.25.3, u and its neighbor v of degree 3 have two common neighbors w_1 and w_2 . If the face containing u , w_1 and w_2 is a triangle, then u receives $\frac{3}{2}$ from w_1 and $\frac{3}{2}$ from w_2 by rule (R_3) . Otherwise, this face gives 1 to u by (R_f) and each w_i , for $i \in \{1, 2\}$, gives 1 to u by (R_3) . In both cases u has final weight 0.
- If $d(u) = 4$, then u must receive at least 2. If $a \geq 2$, then we are done by (R_f) . If $a = 0$, then $w = 4$ and u has final weight at least $-2 + (4 - 1) \times \frac{2}{3} = 0$ by (R_4) since u has at most one 6^- -neighbor, which is non-negative. If $a = 1$, then $t = 3$, $w = 2$ and $s = 2$. In this case, u receives 1 from its non triangular incident face by (R_f) . The vertex u has at most one 4-neighbor by Lemma 9.20, thus u receives at least $\frac{2}{3} + 2 \times \frac{1}{6} = 1$ by (R_4) from its other neighbors, as in the worst case, it has a weak 4-neighbor. Therefore, in all cases, u has non-negative weight.
- If $d(u) = 5$, then u must receive at least 1. If $a \geq 1$, then we are done by (R_f) . If $a = 0$, then $w = 5$ and we also have a non-negative weight by (R_5) as u has at most one 6^- -neighbor by Lemma 9.20.
- If $d(u) = 6$, then u has initial weight 0 and gives/receives nothing.

9.5.3.2 Introducing a tool to simplify the case analysis

In order to simplify the analysis of the cases, we will introduce a notion of *configuration*. A configuration is a set of neighbors of u , all receiving weight from u , and we require for configurations with multiple vertices that the graph induced by these vertices and u is a face of \mathcal{M} . One can see a configuration as a set of 5^- -neighbors of u which are close to each other in \mathcal{M} . All different configurations in an unreducible graph are described in Table 9.1. Some configurations are ignored since they cannot appear around a vertex u with a negative final weight, we describe two of them in Claim 9.44 below. We only consider maximal configurations *i.e.* we associate neighbors of u when possible. Moreover we can always partition the set of neighbors of u which receive weight from u into a set of configurations. For example a weak 3-neighbor of u is a configuration $3w$ and two

adjacent weak 4-neighbors v_1, v_2 of u such that v_1v_2u is a face of \mathcal{M} form a configuration $4w-4w$. Another example is given in Figure 9.16.

We can associate a *weight* $w(c)$ with each configuration c as the sum of the weights given by u to the vertices of c . The total weight given by u by the discharging rules is equal to the sum of the weights of each configuration of u .

Now, suppose that we have a way to count the number of configurations around u , we would be able to bound the maximal weight given by u in the discharging procedure. A naive way to do this is to remark that the sum of the number of vertices in the configurations around u cannot exceed the degree of u but this is too rough to be used in the case analysis.

We define the *degree* $d(c)$ of a configuration c as the number of vertices of the configuration plus half of the number of triangular faces uvz such that v belongs to the configuration and z does not. For example, the degree of the configuration $3sw$ which contains one semi-weak 3-neighbor of u is $1 + \frac{1}{2} = \frac{3}{2}$. The degree of each configuration is given in Table 9.1. One can see an example in Figure 9.16. The following claim helps us to bound the number of configurations around u .

Claim 9.42. *If u has t configurations c_1, \dots, c_t , then $\sum_i d(c_i) \leq d(u)$.*

Proof. We claim that each neighbor of u contributes to at most 1 to the sum of the degrees of the configurations, as there are $d(u)$, we get the conclusion from this statement. Remark that if z is a neighbor of u , then z can belong to at most two triangular faces with u . The vertices of the configuration c contribute to 1 of the degree of c and the vertices z such that uvz is a triangular face, $v \in c$ and $z \notin c$, contribute to $\frac{1}{2}$ of the degree of c . Hence, vertices which are not in a configuration contribute to at most 1 to the sum of the degrees of the configurations.

Suppose uvz is a triangular face such that v belongs to a configuration c and z does not, then z does not receive weight from u since we only consider maximal configurations. Hence a vertex in a configuration cannot contribute $\frac{1}{2}$ to the degree of another configuration. Therefore, each vertex contributes to at most 1 of the degrees of the configurations around u , we get the conclusion. \square

Finally we define the *density* $\rho(c)$ of a configuration c as the ratio $\frac{w(c)}{d(c)}$. The density of each configuration is given in Table 9.1. Let us see how to use the density to bound the total given weight of a vertex. Suppose, for example, that $d(u) = 11$ and that u has exactly one 1-neighbor, no 2-neighbor, exactly two non-adjacent weak 3-neighbors and no other 3^- -neighbor. This means that these configurations $\{1, 3w, 3w\}$ receive a weight of 3 from u and have total degree 5. What is the maximal weight that u could give in this case? A first upper bound is the following: of all other configurations that can appear around u , the $4w-4w$ is the one with the largest density, as we have no other 3^- -neighbor, its density is $\frac{4}{9}$. Thus u gives at most $3 + \frac{4}{9} \times (11 - 5)$ of weight. Indeed suppose c_1, \dots, c_t are the other configurations around u . Then the weight given by u is $3 + \sum_i w(c_i)$. And, we have:

$$\begin{aligned} 3 + \sum_i w(c_i) &\leq 3 + \sum_i \rho(c_i)d(c_i) \\ &\leq 3 + \frac{4}{9} \sum_i d(c_i) \\ &\leq 3 + \frac{4}{9}(d(u) - 5). \end{aligned}$$

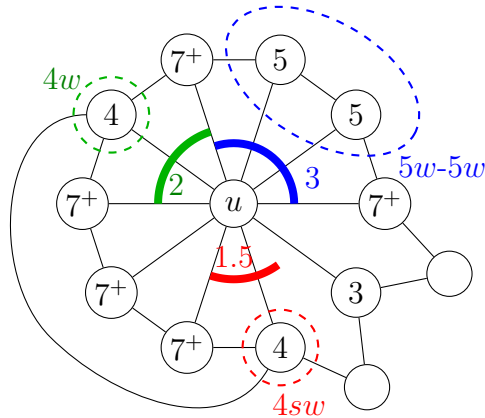


Figure 9.16: The vertex u has three configurations: $4w$, $4sw$ and $5w-5w$. Each configuration is circled by a dashed ellipse. Each arc set around u represents the degree of the corresponding configurations. The number of “sectors” they span is the degree of the configuration and corresponds to the number next to the arc.

Name	Weight	Degree	Density	Description
1	1	1	1	A 1-neighbor
2	1	1	1	A 2-neighbor
$3sw$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{3}$	A semi-weak 3-neighbor with no adjacent 3-vertex
$3w$	1	2	$\frac{1}{2}$	A weak 3-neighbor with no adjacent 3-vertex
$4sw$	$\frac{1}{6}$	$\frac{3}{2}$	$\frac{1}{9}$	A semi-weak 4-neighbor
$4w$	$\frac{2}{3}$	2	$\frac{1}{3}$	A weak 4-neighbor
$4sw-4sw$	$\frac{1}{3}$	2	$\frac{1}{6}$	Two adjacent semi-weak 4-neighbors
$4sw-4w$	$\frac{5}{6}$	$\frac{5}{2}$	$\frac{1}{3}$	Two adjacent 4-neighbors, one weak and one semi-weak
$4w-4w$	$\frac{4}{3}$	3	$\frac{4}{9}$	Two adjacent weak 4-neighbors
$5w$	$\frac{1}{4}$	2	$\frac{1}{8}$	A weak 5-neighbor
$5w-5w$	$\frac{1}{2}$	3	$\frac{1}{6}$	Two adjacent weak 5-neighbors

Table 9.1: All possible configurations in the neighborhood of u . In the case of $4sw$, $4w$, $5sw$ and $5w$, we suppose that the corresponding vertices are not part of a larger configuration $4sw-4sw$, $4sw-4w$, $4w-4w$ or $5w-5w$.

Hence the conclusion. We can do a better analysis by remarking that the $4w-4w$ configuration appears at most once by Lemma 9.25.10. Thus the weight given by u is at most $3 + \frac{4}{3} + \frac{1}{3} \times (11 - 5 - 3)$ as $\frac{1}{3}$ is the largest density among the remaining allowed configurations. Indeed, if the configurations are sorted by decreasing density and j is the smallest index for which $\sum_{i=1}^j d(c_i) \geq 3$, then:

$$\begin{aligned} 3 + \sum_{i=1}^t w(c_i) &\leq 3 + \sum_{i=1}^j \rho(c_i)d(c_i) + \sum_{i=j+1}^t \rho(c_i)d(c_i) \\ &\leq 3 + \frac{4}{9} \times 3 + \frac{1}{3} \sum_{i=j+1}^t d(c_i) \\ &\leq 3 + \frac{4}{3} + \frac{1}{3}(d(u) - 5 - 3). \end{aligned}$$

This type of analysis will greatly decrease the number of cases as a simple computation gives a good enough upper bound in most cases. To get a similar result without these notions would require to analyse all possible cases in the neighborhood of u .

By Lemma 9.25.3 and Lemma 9.25.4, we have the following observation.

Observation 9.43. *If v is a 3-neighbor of u with $c(v) = 1$, then its 3-neighbor w is also a neighbor of u . Moreover, v and w cannot be both weak neighbors of u .*

Let us continue with a small claim.

Claim 9.44. *If u is a 7^+ -vertex with negative final weight, then u does not have two adjacent 3-neighbors.*

Proof. By Lemma 9.25.7, such a vertex u has degree at least 10. By Lemma 9.25.5 and Lemma 9.25.6, u has no other 3^- -neighbor and no weak 4-neighbor. In this case, with Observation 9.43, u gives in total at most $\frac{5}{2} + \frac{1}{6} \times (d(u) - \frac{5}{2})$ as $\frac{1}{6}$ is the largest density available for the other configurations. But u has weight $d(u) - 6$. If u has negative final weight, then by solving the inequality, we obtain $d(u) < 10$, a contradiction. \square

In particular, this claim implies that we can ignore the configurations where v is a 3-neighbor of u and $c(v) = 1$. Indeed, in this case, we know that u has non-negative final weight.

9.5.3.3 Back to the analysis of the discharging

It is left to check the cases where $d(u) \geq 7$. In these cases, u receives no weight but gives some to other vertices. Note that u starts with a weight of $d(u) - 6 \geq 0$. Let us prove that their final weight is still non-negative. Remark that $c(v_i) \leq 1$ when v_i is a 5^- -vertex by Lemma 9.20 and $c(v_i) = 0$ when v_i is a 2^- -vertex by Lemma 9.25.1.

From now on, suppose by contradiction that u has negative weight. In particular with Claim 9.44, we can suppose that every 3-neighbor v of u verifies $c(v) = 0$.

Note that u has at most $\lceil \frac{2}{3}d(u) \rceil$ weak 5^- -neighbors. Indeed, assume this is not the case, by ordering the neighbors of u in a clockwise order around u in \mathcal{M} , there must be at least three consecutive weak 5^- -neighbors of u in this order. In this case, they form a path which contradicts Lemma 9.20.

- If $d(u) = 7$, then u has initial weight 1. If u has at most one 5^- neighbor, then its maximal given weight is 1 and u has a non-negative final weight, a contradiction.

If u has a 4^- -neighbor v_1 and a 5^- -neighbor v_2 , then $d(v_1) + c(v_1) \leq 5$ and $d(v_2) + c(v_2) \leq 6$ by Lemma 9.20. In order to apply Lemma 9.21, we have $m \leq \max(5 - 1, 6 - 2) \leq 4$ and $\binom{k+1-d(u)+p-m}{p} \geq \binom{12+1-7+2-4}{2} = 6$. Note that $d_7(u)+1 \leq 5 + 1 = 6$. Using Lemma 9.21, we can reduce G , a contradiction.

Thus u has only 5^+ -neighbors. If u has at most one configuration $5w-5w$, then its maximal given weight is $\frac{1}{2} + \frac{1}{8} \times (7 - 3) \leq 1$ as $\frac{1}{8}$ is the density of $5w$, the only configuration left. Otherwise, the worst case is to have two $5w-5w$ configurations but in this case, u gives 1. Indeed u cannot have any more weak 5 -neighbor than these four. In both cases, u has non-negative weight, a contradiction.

- If $d(u) = 8$, then u has initial weight 2. If u has at most two 5^- neighbors, then its maximal given weight is 2 and u has a non-negative final weight, a contradiction.

If u has a 4^- -neighbor and three 5^- neighbors, then we can apply Lemma 9.21 as $m \leq 4$, $p = 4$ and $\binom{5}{4} = 5 \geq d_8(u) + 1$, a contradiction.

Suppose u has a 3^- -neighbor. If u has a 3^- -neighbor, a 4^- neighbor and a 5^- neighbor, then we can apply Lemma 9.21 with $m \leq 3$ and $p = 3$ as $\binom{5}{3} = 10 \geq d_8(u)+1$, a contradiction. Thus, by the previous remark, u has no other 4^- -neighbor. It has at most two 5 -neighbors by the second remark. Thus u gives at most $1 + 2 \times \frac{1}{4}$ by (R_1) , (R_2) and (R_5) and therefore has non-negative final weight. Hence u has no 3^- -neighbor.

If u has a 4 -neighbor, then u can have only two other 5^- -neighbors by the second remark. In this case, the maximal given weight is $3 \times \frac{2}{3} = 2$ by (R_4) . Hence u has a non-negative final weight, a contradiction.

Thus u has only 5 -neighbors and u gives at most $8 \times \frac{1}{4} = 2$ by (R_5) , again ending with a non-negative weight, a contradiction.

- If $d(u) = 9$, then u has initial weight 3. If u has at most three 5^- -neighbors, then it has non-negative final weight, a contradiction.

Suppose that u has two 3^- -neighbors. If u has two 3^- -neighbors and a 4^- -neighbor, then with $m \leq 2$ and $p = 3$, we can apply Lemma 9.21 as $\binom{5}{3} = 10 \geq d_9(u) + 1$, a contradiction. If u has at most four 5 -neighbors, then u gives at most $1 + 1 + 4 \times \frac{1}{4} = 3$ by (R_1) , (R_2) and (R_5) and u has a non-negative weight. Then u has at least five 5 -neighbors. We can apply Lemma 9.21 with $m \leq 3$, $p = 5$ by taking the two 3^- -vertices and three of the 5 -vertices as $\binom{6}{5} \geq d_9(u) + 1$, a contradiction.

Hence u has at most one 3^- -neighbor. If u has at most three other 5^- -vertices, then it has non-negative final weight as each of them receives at most $\frac{2}{3}$ by (R_4) and (R_5) .

Suppose that u has a 3^- -neighbor. If u has a 3^- -neighbor and a 4^- -neighbor, then by considering these two vertices and three other 5^- -neighbors, we can apply Lemma 9.21 with $m \leq 3$ and $p = 5$, a contradiction. Thus u has no other 4^- -neighbor and gives at most $1 + 8 \times \frac{1}{4} = 3$ by (R_1) , (R_2) and (R_5) . Thus u does not have a 3^- -neighbor.

Thus u only has 4^+ -neighbors. If it has at most four 5^- -neighbors, then u gives at most $\frac{2}{3} \times 4 < 3$ by (R_4) and (R_5) . By Lemma 9.25.11, two weak 4 -neighbors of u

cannot be adjacent as $d(u) = 9$ and $d_9(u) \leq 4$. Then u gives at most $\frac{1}{3} \times 9 = 3$, as $\frac{1}{3}$ is the largest density among the allowed configurations. Thus u has non-negative weight, a contradiction.

- If $d(u) = 10$, then u has initial weight 4. As we supposed that the final weight of u is negative, u must have at least five 5^- -neighbors.

Suppose that u has two 3^- -neighbors. If u has two 3^- -neighbors and a 4^- -neighbor, then as it has at least two other 5^- -neighbors, we can apply Lemma 9.21 with $m \leq 2$ and $p = 5$, a contradiction. If u has two 3^- -neighbors and no 4^- -neighbor, then u needs at least eleven 5^- -neighbors to be negative by (R_5) , a contradiction with the degree of u . Thus u has at most one 3^- -neighbor.

If u has one 2^- -neighbor, then by Lemma 9.25.9, u cannot have two weak adjacent 4^- -neighbors. The largest density remaining is $\frac{1}{3}$ as it cannot have a 3^- -neighbor. Thus u gives at most $1 + \frac{1}{3} \times (10 - 1) = 4$. Thus it has a non-negative weight, a contradiction.

Suppose u has one 3^- -neighbor. It can have at most one $4w-4w$ configuration by Lemma 9.25.10. The largest density left is $\frac{1}{3}$. Thus u gives at most $1 + \frac{4}{3} + \frac{1}{3} \times (10 - 2 - 3) = 4$. Thus u has non-negative final weight, a contradiction.

If u has no 3^- -neighbor, then the same argument works as u gives at most $\frac{4}{3} + \frac{1}{3} \times (10 - 3) \leq 4$ to its neighbors, a contradiction.

- If $d(u) = 11$, then u has initial weight 5. As we supposed that the final weight of u is negative, u must have at least six 5^- -neighbors.

Suppose that u has two 1^- -neighbors. If u has two 1^- -neighbors and a 3^- -neighbor, then Lemma 9.21 allows us to reduce G with $m = 0$ and $p = 3$, a contradiction. If u has two 1^- -neighbors and no other 3^- -neighbor, then after the first two 1^- -neighbors, the largest density left is $\frac{1}{3}$ by Lemma 9.25.9. Thus u gives at most $1 + 1 + \frac{1}{3} \times (11 - 1 - 1) = 5$. Therefore, u has non-negative weight, a contradiction. Thus u does not have two 1^- -neighbors.

If u has two 2^- -neighbors, then one of them is a 2^- -vertex by the previous argument and Lemma 9.23.1 or Lemma 9.25.2 allows us to reduce, a contradiction.

Suppose u has a 2^- -neighbor. If u has a 2^- -neighbor, two 3^- -neighbors, a 4^- -neighbor and two 5^- -neighbors, then Lemma 9.21 allows us to reduce with $m = 1$ and $p = 6$, a contradiction. If u has a 2^- -neighbor, two 3^- -neighbors, no other 4^- -neighbor, then u gives at most $1 + 1 + 1 + \frac{1}{6} \times (11 - 1 - 2 - 2) = 4 \leq 5$. Thus u has non-negative weight, a contradiction. If u has a 2^- -neighbor and at most one other 3^- -neighbor, then u gives at most $1 + 1 + \frac{1}{3} \times (11 - 1 - 2) = \frac{14}{3} \leq 5$, as the largest density is $\frac{1}{3}$ by Lemma 9.25.9. Thus u has non-negative weight, a contradiction.

If u has no 2^- -neighbor, then u can have at most two weak 3^- -neighbors by Lemma 9.25.8. Thus u gives at most $1 + 1 + \frac{4}{3} + \frac{1}{3} \times (11 - 2 - 2 - 3) = \frac{14}{3} \leq 5$ by Lemma 9.25.10. Therefore, u has non-negative weight, a contradiction.

- If $d(u) \geq 12$, then u has weight $d(u) - 6 \geq \frac{d(u)}{2}$.

If u has no 2^- -neighbors, then the largest density remaining is $\frac{1}{2}$ thus u gives at most $\frac{1}{2} \times d(u) \leq d(u) - 6$ and has a non-negative final weight, a contradiction.

If u has no neighbors v such that $2 \leq d(v) \leq 5$, then Lemma 9.23.2 implies that u has non-negative final weight since it gives at most $d(u) - \left\lceil \frac{k}{2} \right\rceil \leq \frac{d(u)}{2}$.

If it has five 1-neighbors, then there exists at least one vertex of degree at least 2 and at most 5 in the neighborhood of u by the previous remark and Lemma 9.22 raises a contradiction.

Suppose u has four 1-neighbors. If it has four 1-neighbors and a vertex of degree at least 2 and at most 4, then Lemma 9.22 applies again. If it has four 1-neighbors, then u gives at most $4 \times 1 + \frac{1}{6} \times (d(u) - 4) \leq d(u) - 6$ as its other neighbors have degree at least 5. Thus u has at most three 1-neighbors.

If it has two 1-neighbors and a 3-neighbor, then Lemma 9.22 concludes. Note that if u has a 1-neighbor, then it has no 2-neighbor by Lemma 9.23.1. By combining the previous remarks with Lemma 9.25.8, if u has a 1-neighbor, then it has at most two other 3^- -neighbors receiving exactly 1 from u (either 1-vertices or weak 3-neighbors). Then u gives at most $1 + 1 + 1 + \frac{1}{3} \times (d(u) - 3) \leq d(u) - 6$ by Lemma 9.25.9, a contradiction.

Thus u has no 1-neighbor.

Suppose u has a 2-neighbor. By Lemma 9.25.2, u has at most one 2-neighbor. Thus, as in the previous case, by Lemma 9.25.8 and Lemma 9.25.9, u gives at most $1 + 1 + 1 + \frac{1}{3} \times (d(u) - 3) \leq d(u) - 6$, a contradiction.

Thus we can apply Observation 9.41 and this concludes the proof of Lemma 9.18. \square

Proof of Theorem 9.14. Let G be a connected planar graph with $\Delta(G) \geq 12$. Let $k = \Delta(G)$. By Lemma 9.18, G is P-reducible to G_\emptyset thus G is AVD $(\Delta(G) + 1)$ -colorable. \square

9.6 Perspectives

As mentioned previously, it might be possible to generalize Theorem 9.13 to other coloring notions. The key ingredient in this proof is to be able to remove “small vertices” adjacent to other “small vertices”. In particular, the notion of a “small vertex” has some leeway in the way it is defined.

It would also be interesting to lower the lower bound on $\Delta(G)$ in Theorem 9.14. In particular, for $\Delta(G) = 11$, Lemma 9.20 fails to remove 6-vertices adjacent to 5^- -vertices which would be a problem with our current discharging proof. Note that it may be possible to refine the analysis of the proof of Lemma 9.20 to get a more precise statement when the differences between the degrees of the two vertices is at least 2, possibly removing 6-vertices adjacent to 4-vertices. Nonetheless, 5-vertices adjacent to 6-vertices would still pose problems and would need to be dealt with in some other ways.

It would also be interesting to show that $\Delta(G) + 1$ colors are sufficient for some other classes of graphs: *e.g.* graphs with bounded maximum average degree or graphs with bounded maximum degree.

Chapter 10

The Multiplicative 1-2-3 Conjecture

In this chapter, we are particularly interested in the so-called 1-2-3 Conjecture, which is defined through the following notions.

Definition 10.1 (*s*-proper labelling). Given a labelling ℓ (*i.e.* a not necessarily proper edge-coloring) of a graph G , we can compute for every vertex v its *sum* $\sigma_\ell(v)$ of incident labels (*i.e.* edge-colors), being formally $\sigma_\ell(v) = \sum_{u \in N(v)} \ell(uv)$. We say that ℓ is *s-proper* if σ_ℓ yields a proper vertex coloring of G , *i.e.* $\sigma_\ell(u) \neq \sigma_\ell(v)$ for every edge uv . We denote by $\chi_\Sigma(G)$ the smallest $k \geq 1$ such that G admits an *s-proper* k -labelling.

The parameter χ_Σ is the local version of another famous parameter, the *irregularity strength* of a graph, introduced by Chartrand, Jacobson, Lehel, Oellermann, Ruiz and Saba in 1988 [42]. The irregularity strength of a graph G is defined as the smallest integer k for which there exists a k -labelling of G such that no two vertices of the graph (not necessarily adjacent) have the same sum of incident labels. For such a labelling ℓ , replacing each edge e of G by $\ell(G)$ parallel edges creates a multi-graph which is *irregular*, that is to say where no two vertices have the same degree. The corresponding replacing procedure applied with an *s-proper* labelling creates a multi-graph where no two adjacent vertices have the same degree.

Greedy arguments show that there exists only one connected graph G for which $\chi_\Sigma(G)$ is not defined, and that graph is K_2 . This implies that $\chi_\Sigma(G)$ is defined for every graph G with no component isomorphic to K_2 , which we call a *nice graph*. It is then legitimate to wonder how large can $\chi_\Sigma(G)$ be in general, for a nice graph G . Karoński, Łuczak and Thomason [125] conjectured that this value cannot exceed 3 in general.

Conjecture 10.2 (1-2-3 Conjecture [125]). *If G is a nice graph, then $\chi_\Sigma(G) \leq 3$.*

Several aspects towards this conjecture have been investigated to date. For an in-depth review of most of our knowledge on the problem, we refer the reader to the survey [171] by Seamone.

One could naturally wonder about slight modifications of the 1-2-3 Conjecture, where the aim would be to design labellings ℓ which distinguish adjacent vertices accordingly to a function f that is somewhat close to the sum function σ_ℓ . There actually exist at least two such variants which sound particularly interesting due to their respective subtleties and to some behaviors they share with the original 1-2-3 Conjecture.

The first such variant we consider is the one where adjacent vertices of a graph G are required, by a labelling ℓ , to be distinguished by their *multiset* of incident labels. Recall

that a multiset is a set in which elements can be repeated. This notion is similar to AVD coloring, except that the labels do not have to form a proper edge-coloring of the graph.

Definition 10.3 (m-proper labelling). For a vertex v of G , we denote by $\mu_\ell(v)$ the multiset of labels assigned to the edges incident with v . We say that ℓ is *m-proper* if μ_ℓ is a proper vertex coloring of G . We denote by $\chi_M(G)$ the smallest $k \geq 1$ such that G admits an m-proper k -labelling.

The second such variant is the one where adjacent vertices of G must be, by ℓ , distinguished accordingly to the *product* of their incident labels.

Definition 10.4 (p-proper labelling). Formally, for a vertex v of G , we define $\rho_\ell(v)$ as the product of labels assigned to the edges incident with v . We say that ℓ is *p-proper* if ρ_ℓ is a proper vertex coloring of G . We denote by $\chi_\Pi(G)$ the smallest $k \geq 1$ such that G admits a p-proper k -labelling.

There exist several interesting connections between the previous three series of notions. For instance, it can be easily noted that an s-proper or p-proper labelling is always m-proper. As a result, $\chi_M(G) \leq \min\{\chi_\Sigma(G), \chi_\Pi(G)\}$ holds for every graph G for which the parameters are defined (see below). In general, there is no other systematic relationship between these three notions, though some exist in particular contexts. For instance, s-proper 2-labellings, m-proper 2-labellings and p-proper 2-labellings are equivalent notions in regular graphs [15]. It can be noted that s-proper $\{0, 1\}$ -labellings and p-proper $\{1, 2\}$ -labellings are equivalent notions [148]. Another illustration is that an m-proper k -labelling yields a p-proper $\{l_1, \dots, l_k\}$ -labelling, for any set $\{l_1, \dots, l_k\}$ of k pairwise coprime integers.

Just as for the 1-2-3 Conjecture, one can wonder how large can $\chi_M(G)$ and $\chi_\Pi(G)$ be for a given graph G . As mentioned in Chapter 6, similarly as for s-proper labellings, the only connected graph admitting no m-proper labellings is K_2 . This also holds for p-proper labellings. Thus, the notion of nice graph coincides for the three types of proper labellings. It actually turns out that the straight analogue of the 1-2-3 Conjecture was formulated for each of the two variants, that is, three colors suffice for both variants.

The multiset version of the 1-2-3 Conjecture was introduced by Addario-Berry, Aldred, Dalal and Reed in [1], while the product version was introduced by Skowronek-Kaziów in [174]. By an argument above, recall that the sum version and the product version of the 1-2-3 Conjecture, if true, would actually imply the multiset version. From that angle, the multiset version does appear, at least intuitively, as the most feasible out of the three versions. This is reinforced by unique behaviors of m-proper labellings over s-proper labellings and p-proper labellings. In particular, if ℓ is a labelling of a graph and $\mu_\ell(u) = \mu_\ell(v)$ for any two vertices u and v , then we must have $d(u) = d(v)$.

Regarding the sum version, the best result to date, proved by Kalkowski, Karoński and Pfender in [121], is that $\chi_\Sigma(G) \leq 5$ holds for every nice graph G . The conjecture was verified for all 3-colorable graphs [125]. Regarding 4-chromatic graphs, the conjecture was verified for 4-edge-connected ones [202]. In [164], it was recently shown that $\chi_\Sigma(G) \leq 4$ holds for every nice regular graph G .

Regarding the multiset version, for long the best result, proved by Addario-Berry, Aldred, Dalal and Reed in [1], was that $\chi_M(G) \leq 4$ holds for every nice graph G . A few years ago, a breakthrough result was obtained by Vučković in [186], in which he gave a full proof of the conjecture.

Theorem 10.5 (Multiset 1-2-3 Theorem [186]). *If G is a nice graph, then $\chi_M(G) \leq 3$.*

Regarding the product version, the best results to date were mainly obtained via adaptations of arguments used to provide results towards the sum and multiset versions. Specifically, Skowronek-Kaziów proved in [174] that $\chi_\Pi(G) \leq 4$ holds for every nice graph G . In the same article, she proved the product version of the 1-2-3 Conjecture for 3-colorable graphs.

This chapter is devoted to studying various facets of p-proper labellings. Our main result is a proof of the Multiplicative 1-2-3 Conjecture.

Theorem 10.6 (Multiplicative 1-2-3 Theorem). *If G is a nice graph, then $\chi_\Pi(G) \leq 3$.*

This work is joint work with Julien Bensmail, Hervé Hocquard and Éric Sopena. See [17, 18, 19] for the arXiv versions of the three papers related to this chapter.

Our main intention in this chapter is to investigate how the mechanisms in the proof of Theorem 10.5 can be used in the product setting. Our proof of the Multiplicative 1-2-3 Conjecture is presented in Section 10.1.

The Multiplicative 1-2-3 conjecture is tight since some graphs do not admit p-proper 2-labellings. Nonetheless, we aim to see how far 2-labellings are from achieving p-properness. In Section 10.2, our goal is to create 2-labellings which are “almost” p-proper. This leads us to raising a conjecture on almost p-proper 2-labellings, that matches an existing weakening of the sum version of the 1-2-3 Conjecture from [82], which we verify for several classes of graphs.

Another way to improve upon the Multiplicative 1-2-3 Conjecture is to consider its list variant. The list variant of the conjecture is a generalisation where edges must be assigned labels from fixed-size lists that might be different from $\{1, 2, 3\}$. Section 10.3 stands as a preliminary section in which we raise first observations. In that section, we also explore the connections between our list problem and the List 1-2-3 Conjecture, from which we get first systematic upper bounds on the minimum size of the lists for which a p-proper labelling exists. Through the use of the Combinatorial Nullstellensatz, we improve some of these first bounds. These improved bounds are gathered in Section 10.4, and are about both general graphs (Subsection 10.4.1) and particular classes of graphs, such as trees, planar graphs with large girth, and subcubic graphs (Subsection 10.4.2).

Contents

10.1 Proof of the product 1-2-3 Conjecture	267
10.2 A conjecture for 2-labellings with restricted product conflicts	280
10.3 On the list variant of p-proper labellings	282
10.3.1 Early remarks on the parameter ch_Π	283
10.3.2 Connections with the sum variant, and first bounds on ch_Π . .	285
10.3.3 Algebraic tools	286
10.4 Improved bounds on ch_Π^* for some graph classes	287
10.4.1 General graphs	288
10.4.2 Particular classes of graphs	289
10.4.2.1 Paths and cycles	289
10.4.2.2 Trees	291

10.4.2.3 Planar graphs with large girth	294
10.4.2.4 Subcubic graphs	299
10.5 Perspectives	306

10.1 Proof of the product 1-2-3 Conjecture

Let us start by introducing some terminology and recalling some properties of p-proper labellings, which will be used throughout the proof. Let G be a graph, and ℓ be a 3-labelling of G . For a vertex $v \in V(G)$ and a label $i \in \{1, 2, 3\}$, we denote by $d_i(v)$ the i -degree of v by ℓ , being the number of edges incident with v that are assigned label i by ℓ . Note then that $\rho_\ell(v) = 2^{d_2(v)}3^{d_3(v)}$. We say that v is 1-monochromatic if $d_2(v) = d_3(v) = 0$, while we say that v is 2-monochromatic (resp. 3-monochromatic) if $d_2(v) > 0$ and $d_3(v) = 0$ (resp. $d_3(v) > 0$ and $d_2(v) = 0$). In case v has both 2-degree and 3-degree at least 1, we say that v is bichromatic. We also define the $\{2, 3\}$ -degree of v as the sum $d_2(v) + d_3(v)$ of its 2-degree and its 3-degree. Thus, if v is bichromatic, then its $\{2, 3\}$ -degree is at least 2.

Because ℓ assigns labels 1, 2, 3, and, in particular, because 2 and 3 are coprime, note that, for every edge uv of G , we have $\rho_\ell(u) \neq \rho_\ell(v)$ as soon as u and v have different 2-degrees, 3-degrees, or $\{2, 3\}$ -degrees. In particular, u and v cannot be in conflict, *i.e.*, verify $\rho_\ell(u) = \rho_\ell(v)$, if u and v are i -monochromatic and j -monochromatic, respectively, for $i \neq j$, or if u is monochromatic while v is bichromatic.

This property makes labels 2 and 3 very close in terms of vertex multisets and vertex products, since also $\mu_\ell(u) \neq \mu_\ell(v)$ holds as soon as $d_2(u) \neq d_2(v)$ or $d_3(u) \neq d_3(v)$. Thus, the difference between m-proper 3-labellings and p-proper 3-labellings only lies in the behavior of label 1: for the first objects, every edge uv labelled 1 contributes to both $\mu_\ell(u)$ and $\mu_\ell(v)$, while, for the second objects, every edge uv labelled 1 contributes to none of $\rho_\ell(u)$ and $\rho_\ell(v)$. For that reason, m-proper 3-labellings are not p-proper in general; however, there are contexts where this is the case, such as the following meaningful one:

Observation 10.7. *Nice regular graphs verify the product version of the 1-2-3 Conjecture.*

Proof. Let G be a nice k -regular graph. By Theorem 10.5, there exists an m-proper 3-labelling ℓ of G . We claim it is also p-proper. Indeed, by arguments above, if $\rho_\ell(u) = \rho_\ell(v)$ holds for some edge uv , then $d_2(u) = d_2(v)$ and $d_3(u) = d_3(v)$. Since $d(u) = d(v) = k$, this means also $d_1(u) = d_1(v)$ holds. We then deduce that $\mu_\ell(u) = \mu_\ell(v)$ holds, a contradiction. Thus, no two adjacent vertices of G have the same product of labels. \square

Before going into the more general proof of Theorem 10.6, let us start by giving an overview of it. Let G be a nice graph. Our goal is to build a p-proper 3-labelling ℓ of G . We can clearly assume that G is connected. We also set $t = \chi(G)$. Without loss of generality, we can suppose that $t \geq 2$ (and even $t \geq 4$ by [174]).

In what follows, we construct ℓ through three main steps. First, we need to partition the vertices of G in a way verifying specific cut properties, forming what we call a valid partition of $V(G)$ (see later Definition 10.8 for a more formal definition). A *valid partition* $\mathcal{V} = (V_1, \dots, V_t)$ is a partition of $V(G)$ into t independent sets V_1, \dots, V_t fulfilling two main properties, being, roughly put, that 1) every vertex v in some part V_i with $i > 1$ has an incident *upward edge* to every part V_j with $j < i$, and 2) for every connected component

of $G[V_1 \cup V_2]$ having only one edge, we can freely swap its two vertices in V_1 and V_2 while preserving the properties of a valid partition.

Once we have this valid partition \mathcal{V} in hand, we can then start constructing ℓ . The main part of the labelling process, Step 2 below, consists in starting from all edges of G being assigned label 1 by ℓ , and then processing the vertices of V_3, \dots, V_t one after another, possibly changing the labels by ℓ assigned to some of their incident edges, so that certain product types are achieved by ρ_ℓ . These desired product types can be achieved due to the many upward edges that some vertices are incident with (in particular, the deeper a vertex lies in \mathcal{V} , the more upward edges it is incident with). The product types we achieve for the vertices depend on the part V_i of \mathcal{V} they belong to. In particular, the modifications we make on ℓ guarantee that all vertices in V_3, \dots, V_t are bichromatic, every two vertices in V_i and V_j with $i, j \in \{3, \dots, t\}$ and $i \neq j$ have different 2-degrees or 3-degrees, all vertices in V_2 are 1-monochromatic or 2-monochromatic, and all vertices in V_1 are 1-monochromatic or 3-monochromatic. By itself, achieving these product types makes ℓ almost p-proper, in the sense that the only possible conflicts are between 1-monochromatic vertices in V_1 and V_2 . An important point also, is that, through these label modifications, we will make sure that all edges of $G[V_1 \cup V_2]$ remain assigned label 1, and no vertex in $V_3 \cup \dots \cup V_t$ has 3-degree 1, 2-degree at least 2, and odd $\{2, 3\}$ -degree; in last Step 3 below, we will use that last fact to remove remaining conflicts by allowing some vertices of $V_1 \cup V_2$ to become *special*, *i.e.*, make their product realising these exact label conditions.

Step 3 is designed to get rid of the last conflicts between the adjacent 1-monochromatic vertices of V_1 and V_2 without introducing new ones in G . To that end, we will consider the set \mathcal{H} of the connected components of $G[V_1 \cup V_2]$ having conflicting vertices, and, if needed, modify the labels assigned by ℓ to some of their incident edges so that no conflicts remain, and no new conflicts are created in G . To make sure that no new conflicts are created between vertices in $V_1 \cup V_2$ and vertices in $V_3 \cup \dots \cup V_t$, we will modify labels while making sure that all vertices in $V_1 \cup V_2$ are monochromatic or special. An important point also, is that the fixing procedures we introduce require the number of edges in a connected component of \mathcal{H} to be at least 2. Because of that, once Step 2 ends, we must make sure that \mathcal{H} does not contain a connected component with only one edge incident with two 1-monochromatic vertices. To guarantee this, we will also make sure, during Step 2, to modify labels and the partition \mathcal{V} slightly so that \mathcal{H} has no such configuration.

Step 1: Constructing a valid partition

Let $\mathcal{V} = (V_1, \dots, V_t)$ be a partition of $V(G)$ where each V_i is an independent set. Note that such a partition exists, as, for instance, any proper t -coloring of G forms such a partition of $V(G)$. For every vertex $u \in V_i$, an incident *upward edge* (resp. *downward edge*) is an edge uv for which v belongs to some V_j with $j < i$ (resp. $j > i$). Note that all vertices in V_1 have no incident upward edges, while all vertices in V_t have no incident downward edges.

We denote by $M_0(\mathcal{V})$ (also denoted M_0 when the context is clear) the set of isolated edges in the subgraph $G[V_1 \cup V_2]$ of G induced by the vertices of $V_1 \cup V_2$. That is, M_0 contains the edges of the connected components of $G[V_1 \cup V_2]$ that consist in one edge only. To lighten the exposition, whenever referring to the vertices of M_0 , we mean the vertices of G incident with the edges in M_0 .

For an edge $uv \in M_0$ with $u \in V_1$ and $v \in V_2$, *swapping* uv consists in modifying the partition \mathcal{V} by removing u from V_1 (resp. v from V_2) and adding it to V_2 (resp. V_1). In

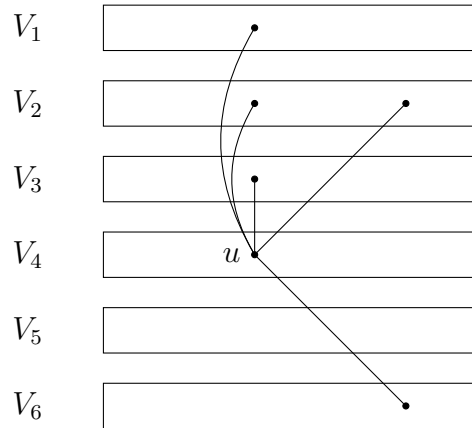


Figure 10.1: The neighborhood of some vertex u of G in a valid partition of G . Note that u belongs to V_4 and has at least one neighbor in each of V_1, V_2 and V_3 . The vertex u can have many other neighbors (here represented by the two other neighbors of u in V_2 and V_6).

other words, we exchange the parts to which u and v belong. Note that if V_1 and V_2 are independent sets before the swap, then, because $uv \in M_0$, by definition the resulting new V_1 and V_2 remain independent. Also, the set M_0 is unchanged by the swap operation.

We can now give a formal definition for the notion of valid partition.

Definition 10.8 (Valid partition). For a t -colorable graph G , a partition $\mathcal{V} = (V_1, \dots, V_t)$ of $V(G)$ is a *valid partition* (of G) if \mathcal{V} verifies the following properties.

- (\mathcal{I}) Every V_i is an independent set.
- (\mathcal{P}_1) Every vertex in some V_i with $i \geq 2$ has a neighbor in V_j for every $j < i$.
- (\mathcal{S}) For every sequence $(e_i)_i$ of edges of $M_0(\mathcal{V})$, successively swapping every e_i (in any order) results in a partition \mathcal{V}' verifying Properties (\mathcal{I}) and (\mathcal{P}_1).

Note that Property (\mathcal{S}) implies the following property:

- (\mathcal{P}_2) Successively swapping any number of edges of $M_0(\mathcal{V})$ results in a valid partition \mathcal{V}' .

See Figure 10.1 for an example of a valid partition.

In order to prove Theorem 10.6, as mentioned earlier, to start constructing ℓ we first need to have a valid partition of G in hand. The following result guarantees its existence.

Lemma 10.9. *Every nice t -colorable graph G admits a valid partition.*

Proof. For a partition $\mathcal{V} = (V_1, \dots, V_t)$ of $V(G)$ where each V_i is an independent set (note that a such partition exists, as every proper t -coloring of G is one such partition), set $f(\mathcal{V}) = \sum_{k=1}^t k \cdot |V_k|$. Among all possible \mathcal{V} 's, we consider a \mathcal{V} that minimises $f(\mathcal{V})$.

Suppose that there is a vertex $u \in V_i$ with $i \geq 2$ for which Property (\mathcal{P}_1) does not hold, *i.e.*, there is a $j < i$ such that u has no incident upward edge to V_j . By moving u to V_j , we obtain another partition \mathcal{V}' of $V(G)$ where every part is an independent set. However, note that $f(\mathcal{V}') = f(\mathcal{V}) + j - i < f(\mathcal{V})$, a contradiction to the minimality of \mathcal{V} . From this, we deduce that every partition \mathcal{V} minimising f must verify Property (\mathcal{P}_1).

Let now \mathcal{V}' be the partition of $V(G)$ obtained by successively swapping edges of $M_0(\mathcal{V})$. Recall that the swapping operation preserves Property (\mathcal{I}) and observe that $f(\mathcal{V}) = f(\mathcal{V}')$. Hence, \mathcal{V}' minimises f and thus verifies Properties (\mathcal{I}) and (\mathcal{P}_1). Thus Property (\mathcal{S}) also holds, and \mathcal{V} is a valid partition of G . \square

From here, we thus assume that we have a valid partition $\mathcal{V} = (V_1, \dots, V_t)$ of G .

Step 2: Labelling the upward edges of V_3, \dots, V_t

From G and \mathcal{V} , our goal now is to construct a 3-labelling ℓ of G achieving certain properties, the most important of which being that the only possible conflicts are between pairs of vertices of V_1 and V_2 that do not form an edge of M_0 . The following result sums up the exact conditions we want ℓ to fulfil. Recall that a vertex v is special by ℓ , if $d_3(v) = 1$, $d_2(v) \geq 2$ and $d_2(v) + d_3(v)$ is odd. Note that special vertices are bichromatic.

Lemma 10.10. *For every nice graph G and every valid partition (V_1, \dots, V_t) of G , there exists a 3-labelling ℓ of G such that:*

1. all vertices of V_1 are either 1-monochromatic or 3-monochromatic,
2. all vertices of V_2 are either 1-monochromatic or 2-monochromatic,
3. all vertices of $V_3 \cup \dots \cup V_t$ are bichromatic,
4. no vertex is special,
5. if $u \in V_1$ and $v \in V_2$ are adjacent, then $\ell(uv) = 1$,
6. if two vertices u and v are in conflict, then $u \in V_1$ and $v \in V_2$ (or vice versa), and at least one of u or v has a neighbor w in $V_1 \cup V_2$.

Proof. From now on, we fix the valid partition $\mathcal{V} = (V_1, \dots, V_t)$ of G . During the construction of ℓ , we may have, however, to swap some edges of M_0 , resulting in a different valid partition of G . Abusing the notations, for simplicity we will still denote by \mathcal{V} any valid partition of G obtained this way, through swapping edges. Recall that valid partitions are closed under swapping edges of M_0 (Property (\mathcal{P}_2) of Definition 10.8).

Our goal is to design ℓ so that it not only verifies the four color properties of Items 1 to 4 of the statement, but also achieves the following refined product types, for every vertex v in a part V_i of \mathcal{V} :

- $v \in V_1$: v is 1-monochromatic or 3-monochromatic,
- $v \in V_2$: v is 1-monochromatic or 2-monochromatic,
- $v \in V_3$: v is bichromatic, has 2-degree 1 and even $\{2, 3\}$ -degree,
- $v \in V_4$: v is bichromatic, has 3-degree 2 and odd $\{2, 3\}$ -degree,
- $v \in V_5$: v is bichromatic, has 2-degree 2 and even $\{2, 3\}$ -degree,
- ...
- $v \in V_{2n}, n \geq 3$: v is bichromatic, has 3-degree n and odd $\{2, 3\}$ -degree,
- $v \in V_{2n+1}, n \geq 3$: bichromatic, has 2-degree n and even $\{2, 3\}$ -degree,
- ...

We start from ℓ assigning label 1 to all edges of G . Let us now describe how to modify ℓ so that the conditions above are met for all vertices. We consider the vertices of V_t, \dots, V_3 following that order, from “bottom to top”, and modify labels assigned to upward edges. An important condition we will maintain, is that every vertex in an odd

part V_{2n+1} ($n \geq 0$) has all its incident downward edges (if any) labelled 3 or 1, while every vertex in an even part V_{2n} ($n \geq 1$) has all its incident downward edges (if any) labelled 2 or 1. Note that this is trivially verified for the vertices in V_t , since they have no incident downward edges.

At any point in the process, let M be the set of edges of M_0 for which both ends are 1-monochromatic (initially, $M = M_0$). When treating a vertex $u \in V_3 \cup \dots \cup V_t$, we define M_u as the subset of edges of M having an end that is a neighbor of u . For every edge $e \in M_u$, we choose one end of e that is a neighbor of u and we add it to a set S_u . Note that $|S_u| = |M_u|$. Another goal during the labelling process, to fulfil Item 6, is to label the edges incident with u so that at least one end of every edge in M_u is no longer 1-monochromatic. Note that the set M_u considered when labelling the edges incident with u is not necessarily the set of edges of M_0 incident with a neighbor of u , as, during the whole process, some of these edges might be removed from M when dealing with previous vertices in $V_3 \cup \dots \cup V_t$.

Let us now consider the vertices in V_t, \dots, V_3 one by one, following that order. Let thus $u \in V_i$ be a vertex that has not been treated yet, with $i \geq 3$. Recall that every vertex belonging to some V_j with $j > i$ was treated earlier on, and thus has its desired product. Suppose that $i = 2n$ with $n \geq 2$ (resp. $i = 2n + 1$ with $n \geq 1$). Recall also that u is assumed to have all its incident downward edges labelled 1 or 2 (resp. 3), due to how vertices in V_j 's with $j > i$ have been treated earlier on. Also, all upward edges incident with u are currently assigned labelled 1 by ℓ .

If $M_u \neq \emptyset$, then we swap edges of M_u , if necessary, so that every vertex in S_u belongs to V_2 (resp. V_1). This does not invalidate any of our invariants since both ends of an edge in S_u are 1-monochromatic.

In any case, by Property (\mathcal{P}_1) , we know that, for every $j < i$, there is a vertex $x_j \in V_j$ which is a neighbor of u . In particular, the vertex x_1 (resp. x_2) does not belong to S_u (but may be the other end of an edge in M_u). We label the edges $ux_3, ux_5, \dots, ux_{2n-1}$ with 3 (resp. $ux_4, ux_6, \dots, ux_{2n}$ with 2). Note that, at this point, $d_3(u) = n - 1$ (resp. $d_2(u) = n - 1$). To finish dealing with u , we need to distinguish two cases depending on whether M_u is empty or not.

- Suppose first that $M_u = \emptyset$. Label ux_1 with 3 (resp. ux_2 with 2). Now u has the desired 3-degree (resp. 2-degree). If $i > 3$, then label ux_{i-2} with 2 (resp. 3) so that u is sure to be bichromatic. If $i > 3$ and the $\{2, 3\}$ -degree of u does not have the desired parity, then label ux_2 with 2 (resp. ux_1 with 3). If $u \in V_3$ and the $\{2, 3\}$ -degree of u is even, then u is already bichromatic since $d_2(u) = 1$. If $u \in V_3$ and the $\{2, 3\}$ -degree of u is odd, then label ux_1 with 3 to adjust the parity of the $\{2, 3\}$ -degree of u and making u bichromatic. In all cases, at this point u is bichromatic with 3-degree n (resp. 2-degree n) and odd $\{2, 3\}$ -degree (resp. even $\{2, 3\}$ -degree), which is precisely what is desired for u .
- Suppose now that $M_u \neq \emptyset$. Let $z \in S_u$ and let e be the edge of M_u containing z . For every vertex $w \in S_u \setminus \{z\}$, we label the edge uw with 2 (resp. 3). Then:
 - If $d_2(u) + d_3(u)$ is odd (resp. even), then label uz with 2 (resp. 3) and ux_1 with 3 (resp. ux_2 with 2). In this case, every edge in M_u is incident with at least one vertex which is not 1-monochromatic, while u is bichromatic with 3-degree n (resp. 2-degree n) and odd $\{2, 3\}$ -degree (resp. even $\{2, 3\}$ -degree).

- If $d_2(u) + d_3(u)$ is even (resp. odd) and $d_2(u) > 0$ (resp. $d_3(u) > 0$), then swap e and label uz with 3 (resp. 2). Note that, after the swap of e , we have $z \in V_1$ (resp. $z \in V_2$). In this case, every edge in M_u is incident with at least one vertex which is not 1-monochromatic, while u is bichromatic with 3-degree n (resp. 2-degree n) and odd $\{2, 3\}$ -degree (resp. even $\{2, 3\}$ -degree).
- The last case is when $d_2(u) + d_3(u)$ is even (resp. odd) and $d_2(u) = 0$ (resp. $d_3(u) = 0$). If $i > 4$, then we can label ux_{i-2} with 2 (resp. 3) and fall back into one of the previous cases. If $i = 4$, then the only edge labelled 3 is the edge ux_3 which implies that $d_3(u) = 1$, which is impossible since $d_2(u) = 0$ (resp. $d_3(u) = 0$) and $d_2(u) + d_3(u)$ is odd. If $i = 3$, then the conditions of this case imply that $d_2(u) = 1$ while every upward edge incident with u is labelled 1 or 3 and similarly for every incident downward edge; this case thus cannot occur.

To finish, we remove the edges of M_u from M since their two ends are not both 1-monochromatic any more.

At the end of this process, all vertices in V_1 are 1-monochromatic or 3-monochromatic, while all vertices in V_2 are 1-monochromatic or 2-monochromatic. Every vertex in $V_3 \cup \dots \cup V_t$ is bichromatic and there are no conflicts involving any pair of these vertices. Indeed if $a \in V_i$ and $b \in V_j$ are adjacent with $i > j \geq 3$, then either i and j do not have the same parity, in which case a and b do not have the same $\{2, 3\}$ -degree; or both i and j are even (resp. odd) and $d_3(a) = \frac{i}{2} \neq \frac{j}{2} = d_3(b)$ (resp. $d_2(a) = \frac{i-1}{2} \neq \frac{j-1}{2} = d_2(b)$). Note also that no vertex in G is special, as, by definition, special vertices have 3-degree 1, 2-degree at least 2, and odd $\{2, 3\}$ -degree. Moreover, we did not change the label of any edge in the cut (V_1, V_2) .

Finally, suppose that there is a conflict between two vertices u and v . Previous remarks imply that $u \in V_1$ and $v \in V_2$ (or *vice versa*) and that both u and v are 1-monochromatic. If none of u and v has another neighbor w in $V_1 \cup V_2$, then the edge uv belongs to the set M_0 . Since G is nice, one of u or v must have a neighbor z in $V_3 \cup \dots \cup V_t$. Hence $uv \in M_z$. Recall also that we relabelled the edges incident with z in such a way that, for every edge of M_z , at least one incident vertex became 2-monochromatic or 3-monochromatic, a contradiction to the existence of u and v . Hence, all properties of the lemma hold. \square

Step 3: Labelling the edges between V_1 and V_2

From now on, we will modify a 3-labelling ℓ of G obtained by applying Lemma 10.10. We denote by \mathcal{H} the set of the connected components of $G[V_1 \cup V_2]$ that contain two adjacent vertices $u \in V_1$ and $v \in V_2$ having the same product by ℓ . By Items 1 and 2 of Lemma 10.10, such u and v are 1-monochromatic. Also, by Item 6 of Lemma 10.10, recall that every connected component of \mathcal{H} has at least two edges. In what follows, we only relabel edges of some connected components $H \in \mathcal{H}$ with making sure that their vertices (in $V_1 \cup V_2$) are monochromatic or special. This ensures that only vertices of H have their product affected, thus that no new conflicts involving vertices in $V_3 \cup \dots \cup V_t$ are created.

For a subgraph X of $H \in \mathcal{H}$ (possibly $X = H$), if, after having relabelled edges of X , no conflict remains between vertices of X and all vertices of X are either monochromatic or special, then we say that X verifies Property (\mathcal{P}_3) .

Lemma 10.11. *If we can relabel the edges of every $H \in \mathcal{H}$ so that every H verifies Property (\mathcal{P}_3) , then the resulting 3-labelling is p -proper.*

Proof. This is because if we get rid of all conflicts in \mathcal{H} , then the only possible remaining conflicts are between vertices in $V_1 \cup V_2$ and in $V_3 \cup \dots \cup V_t$. In particular, recall that any two vertices of two distinct connected components $H_1, H_2 \in \mathcal{H}$ cannot be adjacent. Note also that, because we only relabelled edges in \mathcal{H} , the vertices in $V_3 \cup \dots \cup V_t$ retain the product types described in Lemma 10.10. In particular, they remain bichromatic and none of them is special. Thus, they cannot be in conflict with the vertices in $V_1 \cup V_2$. \square

In order to show that we can relabel the edges of every $H \in \mathcal{H}$ so that it fulfils Property (\mathcal{P}_3) , the following result will be particularly handy.

Lemma 10.12. *For every integer $s \in \{2, 3\}$, every connected bipartite graph H whose edges are labelled 1 or s , and any vertex v in any part $V_i \in \{V_1, V_2\}$ of H , we can relabel the edges of H with 1 and s so that $d_s(u)$ is odd (resp. even) for every $u \in V_i \setminus \{v\}$, and $d_s(u)$ is even (resp. odd) for every $u \in V_{3-i}$.*

Proof. As long as H has a vertex u different from v that does not verify the desired condition, apply the following. Choose P any path from u to v , which exists by the connectedness of H . Now follow P from u to v , and change the labels of the traversed edges from 1 to s and *vice versa*. It can be noted that this alters the parity of the s -degrees of u and v , while this does not alter that parity for any of the other vertices of H . Thus, this makes u satisfy the desired condition, while the situation did not change for the other vertices different from u and v . Thus, once this process ends, all vertices of H different from v have their s -degree being as desired by the resulting labelling. \square

We are now ready to treat the connected components $H \in \mathcal{H}$ independently, so that they all meet Property (\mathcal{P}_3) . To ease the reading, we distinguish several cases depending on the types and on the degrees of the vertices that H includes. In each of the successive cases we consider, it is implicitly assumed that H does not meet the conditions of any previous case.

Claim 10.13. *If $H \in \mathcal{H}$ contains a 3-monochromatic vertex $v \in V_1$, or a 1-monochromatic vertex $v_1 \in V_1$ having two 1-monochromatic neighbors $u_1, u_2 \in V_2$ with degree 1 (in H), then we can relabel edges of H so that H verifies Property (\mathcal{P}_3) .*

Proof. Recall that all edges of H (and thus in \mathcal{H}) are assigned label 1; thus, if a vertex of H is 3-monochromatic, then it must be due to incident downward edges with V_3, \dots, V_t .

If H has a 1-monochromatic vertex $v_1 \in V_1$ having two 1-monochromatic neighbors $u_1, u_2 \in V_2$, then we set $\ell(v_1u_1) = \ell(v_1u_2) = 3$. Note that u_1 and u_2 become 3-monochromatic with 3-degree 1, and are thus no longer in conflict with v_1 , as it becomes 3-monochromatic with 3-degree 2. Note that either we got rid of all conflicts in H and H now verifies Property (\mathcal{P}_3) as desired, or conflicts between other 1-monochromatic vertices of H remain. In the latter case, we continue with the following arguments.

Assume H has remaining conflicts, and that H has a 3-monochromatic vertex $v \in V_1$ (and, due to the previous process, perhaps 3-monochromatic vertices u_1 and u_2 in V_2 , in which case their 3-degree (and degree) is precisely 1, while their unique neighbor v in $V_1 \cap V(H)$ is 3-monochromatic with 3-degree 2). Let X be the set of all 3-monochromatic vertices of H belonging to V_1 . Let C_1, \dots, C_q denote the $q \geq 1$ connected components of $H - X$ that do not consist in a 3-monochromatic vertex of V_2 (the vertices u_1 and u_2 we dealt with earlier on). For every C_i , we choose arbitrarily a vertex $x_i \in X$ and a vertex $y_i \in C_i$ such that x_i and y_i are adjacent in H . Note that the vertices of C_i are

either 1-monochromatic or 2-monochromatic (in which case they belong to V_2), since all 3-monochromatic vertices of H are part of X (or are the vertices u_1 and u_2 dealt with earlier on, which we have omitted for now and are not part of the C_i 's).

By Lemma 10.12, in every C_i we can relabel the edges with 1 and 2 so that all vertices in $(V_2 \cap V(C_i)) \setminus \{y_i\}$ are 2-monochromatic with odd 2-degree, while all vertices in $V_1 \cap V(C_i)$ are 2-monochromatic with even 2-degree or possibly 1-monochromatic if their even 2-degree is 0. In particular, recall that y_i must be 1-monochromatic or 2-monochromatic. If y_i has odd 2-degree, then there are no conflicts between vertices of C_i . If y_i has even non-zero 2-degree, then we set $\ell(x_i y_i) = 3$, thereby making y_i special.

Let Y be the set containing all 1-monochromatic y_i 's having a 1-monochromatic neighbor w_i in C_i . Let H' be the subgraph of H induced by $Y \cup X$. Note that every edge of H' is labelled 1. Let now Q_1, \dots, Q_p denote the connected components of H' and choose $x_k \in X \cap V(Q_k)$ for every $k \in \{1, \dots, p\}$. For every k , we apply Lemma 10.12 with labels 1 and 3 so that all vertices in $V_2 \cap V(Q_k)$ get 3-monochromatic with odd 3-degree, while all vertices in $V_1 \cap V(Q_k) \setminus \{x_k\}$ get 3-monochromatic with even 3-degree or possibly 1-monochromatic if their 3-degree is 0.

If x_k is involved in a conflict with a vertex $y_i \in V_2 \cap V(Q_k)$, then this is because x_k has odd 3-degree. Then:

- If $\ell(x_k y_i) = 3$, then $d_3(y_i) = d_3(x_k) \geq 3$ since $x_k \in X$ (x_k must thus be incident with at least one other edge labelled 3, either a downward edge to V_3, \dots, V_t or an edge incident with u_1 (and similarly an edge incident with u_2)). We here assign label 1 to the edge $x_k y_i$ and label 3 to the edge $y_i w_i$. This way, x_k gets even 3-degree while the 3-degree of y_i does not change. Note that y_i and w_i are not in conflict since $d_3(w_i) = 1$ and $d_3(y_i) \geq 3$.
- Otherwise, if $\ell(x_k y_i) = 1$, then we assign label 3 to the edge $x_k y_i$ and label 3 to the edge $y_i w_i$. This way, x_k gets even 3-degree while the 3-degree of y_i remains odd and must be at least 3. Again y_i and w_i are not in conflict since $d_3(w_i) = 1$ and $d_3(y_i) \geq 3$.

We claim that we got rid of all conflicts in H . Indeed, consider two adjacent vertices $a \in V_1 \cap V(H)$ and $b \in V_2 \cap V(H)$. Suppose first that a and b belong to some C_i . Note that, with the exception of y_i and maybe of the vertex w_i (if it exists and $y_i \in Y$), every vertex of C_i is 1-monochromatic or 2-monochromatic, the vertices of $V_1 \cap V(C_i)$ having even 2-degree and the vertices of $V_2 \cap V(C_i)$ having odd 2-degree. Thus, no conflict involves two of these vertices. Suppose now that $b = y_i$. If y_i is 2-monochromatic with odd 2-degree, then there is no conflict involving y_i in C_i since all of its neighbors in C_i have even 2-degree. If y_i is special, then it is the only special vertex of C_i , so, here again, it cannot be involved in a conflict. If $y_i \notin Y$ and y_i is 1-monochromatic, then y_i has no other 1-monochromatic neighbor in C_i by definition of Y . If $y_i \in Y$, then y_i is 3-monochromatic with odd 3-degree, the only other possible 3-monochromatic neighbor of y_i in C_i being w_i , but we showed previously that their 3-degrees differ. Thus, in all cases, there cannot be conflicts between vertices of C_i .

We are left with the case where a and b do not belong to the same C_i . In particular, this implies that $a \in X$ and that a is 3-monochromatic. The only possible 3-monochromatic vertices in V_2 are the vertices of Y , which have odd 3-degree, and the 3-monochromatic vertices u_1 and u_2 with 3-degree 1 and degree 1 in H which might have been created at the very beginning of the proof. If $b \in Y$, then, due to the application of Lemma 10.12 above,

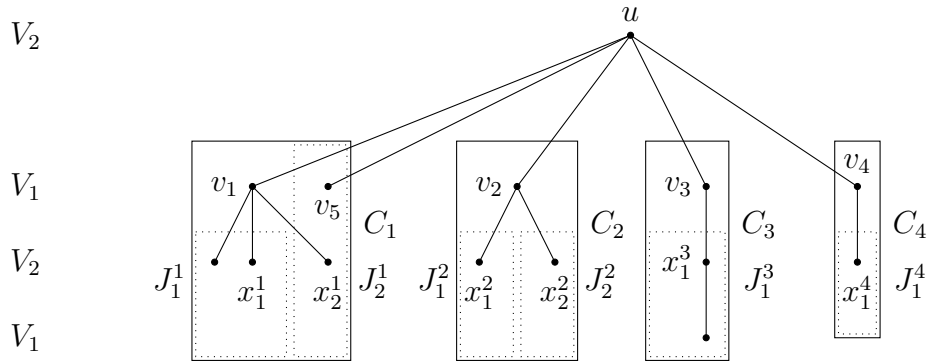


Figure 10.2: An example of the notation of the proof of Claim 10.14. Every rectangular box represents a connected component (in particular every vertex represented inside a box is connected by paths to the others even though not all edges are represented). Boxes with a full border represent the C_i 's while boxes with a dotted border represent the J_j^i 's. Note that if there are no other vertices beside the ones on the figure in both C_3 and C_4 , then C_3 is a tricky component while C_4 is a bad component.

the only vertex of X which can have odd 3-degree is some x_k , but for this vertex we either ensured that it was involved in no conflict, or we tweaked the labelling so that it got even 3-degree without modifying the labelling properties obtained through Lemma 10.12. If b is u_1 or u_2 , then b has only one neighbor v . Note that the edges vu_1 and vu_2 are still labelled 3 as they are not part of the Q_i 's, and, thus, $d_3(b) = 1$ and $d_3(v) \geq 2$. Hence, there is no conflict between vertices of X and other vertices of H . This implies that H verifies Property (\mathcal{P}_3) . \square

Claim 10.14. *If H contains a 1-monochromatic vertex $u \in V_2$ with at least two neighbors in H , then we can relabel edges of H so that H verifies Property (\mathcal{P}_3) .*

Proof. Let v_1, \dots, v_p denote the neighbors of u in H . Due to Lemma 10.10 and because Claim 10.13 does not apply on H , for every vertex v of H we have $d_3(v) = 0$. In particular, none of the v_i 's is 3-monochromatic, implying that they are all 1-monochromatic. Let C_1, \dots, C_q be the $q \geq 1$ connected components of $H - u$. Every C_i contains at least one of the v_i 's. Up to renaming the v_i 's, we can suppose w.l.o.g. that $v_i \in V(C_i)$ if $i \leq q$. The vertices v_i with $i > q$ (if any) can belong to any of the C_i 's.

Let us focus on one component C_i . Let J_1^i, \dots, J_r^i denote the r connected components of $C_i - v_i$. If C_i has order 1, then by convention we set $r = 0$. In every J_j^i , choose a neighbor x_j^i of v_i (see Figure 10.2). By Lemma 10.12, we can relabel edges of J_j^i with 1 and 2 so that every vertex of $V_1 \cap V(J_j^i)$ has even 2-degree, while every vertex of $V_2 \cap V(J_j^i)$, except possibly x_j^i , has odd 2-degree. Let X_i be the set containing all x_j^i 's with even 2-degree. Note that v_i has even 2-degree, being precisely 0 since it is 1-monochromatic; thus the only possible conflicts in C_i involve vertices of X_i as they are the only ones not following the parity rule on their 2-degree (that is, they have even 2-degree).

If $|X_i| = 0$, $|X_i| \geq 2$ or if $X_i = \{w_i\}$ and $d_2(w_i) \geq 1$ for some vertex w_i , then we say that C_i is *nice*. In this case, we can relabel edges of C_i so that C_i verifies Property (\mathcal{P}_3) . If $|X_i| = 0$, then C_i already verifies Property (\mathcal{P}_3) . If $|X_i| \geq 2$, then, for every $z \in X_i$, set $\ell(v_i z) = 3$. If $X_i = \{w_i\}$ and $d_2(w_i) \geq 1$, then set $\ell(v_i w_i) = 3$. In the last two cases, all vertices of X_i either become special while they have no special neighbors; or they become 3-monochromatic with 3-degree 1 in which case v_i is their only 3-monochromatic neighbor

and $d_3(v_i) \geq 2$. Moreover, in both cases, $d_3(v_i) \geq 1$ and all the neighbors of v_i which are not in X_i have 3-degree 0. Thus, v_i cannot be in conflict with its neighbors. Because the products of the other vertices of C_i were not altered by these labelling modifications, C_i verifies Property (\mathcal{P}_3) .

If $X_i = \{w_i\}$ and w_i is 1-monochromatic with no such neighbors in $C_i - v_i$, then we say that C_i is *bad*. In such a bad component C_i , the only current conflict is between v_i and w_i . If $X_i = \{w_i\}$ and w_i is 1-monochromatic with at least one 1-monochromatic neighbor y_i in $C_i - v_i$, then we say that C_i is *tricky*. We denote by N_n the number of nice components, by N_b the number of bad components, and by N_t the number of tricky components. Let also N_{an} be the number of neighbors of u with 2-degree 0 belonging to nice components. Note that $N_{an} \geq N_n$ as, in particular, for every nice component C_i , vertex v_i is a neighbor of u with 2-degree 0.

In what follows, we consider several cases. In each case, we implicitly assume that none of the previous cases applies.

- **Case 1.** $N_t > 0$.

Let C_i be a tricky component. For every bad or tricky component C_j with $j \neq i$, set $\ell(v_j w_j) = 2$ and $\ell(u v_j) = 2$. In C_j , every vertex of V_1 now has even 2-degree since $d_2(v_j) = 2$ and every vertex of V_2 has odd 2-degree since $d_2(w_j) = 1$.

At this point, if $d_2(u)$ is even, then set $\ell(v_i w_i) = 2$ and $\ell(u v_i) = 2$. Here, C_i behaves exactly like the other bad or tricky components and thus contains no conflicts.

If $d_2(u)$ is odd, then set $\ell(v_i w_i) = \ell(w_i y_i) = 3$. Recall that all conflicts of C_i involved w_i . Note that w_i is now 3-monochromatic with 3-degree 2 and no such neighbors. The vertices y_i and v_i are now 3-monochromatic with 3-degree 1 and no such neighbors (in particular, they are not adjacent since they both belong to V_1). Hence C_i does not contain any conflict.

In both cases, note that u is 2-monochromatic with odd 2-degree. To summarise, we have reached the following situation. Special vertices (which were only created when dealing with nice components) only belong to V_2 . 3-monochromatic vertices are involved in no conflicts inside their component C_j and have no 3-monochromatic neighbors outside C_j since $d_3(u) = 0$. All the other vertices of H are either 1-monochromatic or 2-monochromatic: in particular, they have even 2-degree if they belong to V_1 , while they have odd 2-degree if they belong to V_2 . Hence, there is no conflict in H , and H thus verifies Property (\mathcal{P}_3) .

From now on, we can thus suppose that none of the C_i 's is tricky.

- **Case 2.** $N_{an} = N_n = 0$.

Note that we must indeed have $N_{an} = N_n$ when $N_n = 0$. In this case, all C_i 's are bad. If $N_b = 1$, *i.e.*, H contains only one (bad) component C_1 , then set $\ell(v_1 w_1) = 2$ and $\ell(u v_1) = 2$. Then every vertex of H in V_1 is 1-monochromatic or 2-monochromatic with even 2-degree, while every vertex of H in V_2 is 2-monochromatic with odd 2-degree. Indeed $d_2(w_1) = 1$, $d_2(v_i) = 2$ and $d_2(u) = 1$.

If $N_b > 1$, then, for every (bad) component C_i , set $\ell(u v_i) = 3$. Note that this makes all vertices of H be monochromatic. Every neighbor z of u verifies $d_3(z) \leq 1$ and, because $d_3(u) \geq 2$, vertex u cannot be in conflict with any of its neighbors in H .

The vertices v_i with $i \leq q$ are 3-monochromatic with 3-degree 1 and have no such neighbors. The w_i 's are 1-monochromatic and have no 1-monochromatic neighbors since the C_i 's were bad and their v_i 's are no longer 1-monochromatic. The other 1-monochromatic vertices and 2-monochromatic vertices raise no conflicts since, for every such vertex z in $V_j \cap V(H)$ (where $j \in \{1, 2\}$), we have $d_2(z) \equiv j - 1 \pmod{2}$.

Hence H verifies Property (\mathcal{P}_3) in both cases.

- **Case 3.** $N_{an} = 1$.

Recall that this implies $N_n = 1$. Let thus C_1 be the nice component. Suppose first that $H - u$ is connected, *i.e.*, $q = 1$ and C_1 is the only component. As we assumed that $d(u) \geq 2$, vertex u has at least one other neighbor v_2 in C_1 . Since $N_{an} = 1$ and $d_2(v_1) = 0$, vertex v_2 cannot have 2-degree 0, meaning that v_2 has even 2-degree with $d_2(v_2) \geq 2$ (since, in a nice component, every vertex of V_1 has even 2-degree). If v_1 is 3-monochromatic, then there are no conflicts involving u in H . If v_1 is 1-monochromatic, then H contains no 3-monochromatic vertex and no special vertex (due to how we modified labels in nice components). Thus, setting $\ell(uv_2) = 3$ makes u become 3-monochromatic with no such neighbor and v_2 become special with no such neighbor. The 1-monochromatic vertices and 2-monochromatic vertices raise no conflicts, as, for every such vertex z in $V_j \cap V(H)$, $j \in \{1, 2\}$, we have $d_2(z) \equiv j - 1 \pmod{2}$. Hence H verifies Property (\mathcal{P}_3) .

Suppose now that there are at least two connected components in $H - u$. Since $N_t = 0$ and $N_n \leq N_{an} = 1$, all the components different from C_1 must be bad. For every bad component C_j , set $\ell(v_i w_i) = 2$ and $\ell(uv_i) = 2$. In C_j , every vertex of V_1 has even 2-degree since $d_2(v_j) = 2$ and every vertex of V_2 has odd 2-degree since $d_2(w_j) = 1$. Now:

- If $d_2(u)$ is odd, then we claim that we have no conflicts in H . First, we saw earlier that any two vertices of C_1 cannot be in conflict. Next, in every C_i with $i > 1$, every vertex of V_1 has even 2-degree, while every vertex of V_2 has odd 2-degree; hence any two vertices of C_i cannot be in conflict. Thus, every possible conflict in H must involve u . Note that u is 2-monochromatic with odd 2-degree while every vertex of $V_1 \cap V(H)$ has even 2-degree. Thus u cannot be in conflict with a vertex of H , and H verifies Property (\mathcal{P}_3) .
- If $d_2(u)$ is even, then set $\ell(uv_1) = 3$. Again there is no conflict between two vertices in any C_i with $i > 1$. Since only v_1 changed product in C_1 , if there is a conflict between two vertices of C_1 , then it must involve v_1 . Note that v_1 is 3-monochromatic. If $d_3(v_1) \geq 2$, then it is the only vertex of C_1 with this property. If $d_3(v_1) = 1$, then v_1 was 1-monochromatic before the labelling of uv_1 , in which case v_1 , now, has no 3-monochromatic neighbors in C_1 by construction. In both cases, v_1 cannot be in conflict with any other vertex of C_1 . The last possibility for a conflict involves u . Note that u is special and that every special vertex of H belongs to V_2 (more precisely, to the nice component C_1); thus, once again, u cannot be involved in a conflict. Hence H verifies Property (\mathcal{P}_3) .

Thus, in both cases, H verifies Property (\mathcal{P}_3) .

- **Case 4.** $N_{an} \geq 2$.

Thus, $N_n \geq 1$. Let $A = \{a_1, \dots, a_r\}$ be the subset of neighbors of u with 2-degree 0 that are not a v_i for which $i \leq q$ and C_i is a bad component. In other words, the a_i 's are all neighbors of u with 2-degree 0 that are not a v_i with C_i bad. Note that some of these a_i 's are v_i 's (thus in nice components), in which case, by how the nice components were treated earlier, they can be 3-monochromatic. Note also that A may contain more than N_{an} vertices since it may also contain 1-monochromatic v_i 's with 2-degree 0 and $i > q$ and which belong to a bad component.

For every bad component C_i , set $\ell(uv_i) = 3$. Then, for every $a_i \in A$, we define n_i as the value of $d_3(a_i)$ at the beginning of this case. Recall that we can have $d_3(a_i) > 0$, in case a_i is a v_j for which C_j is nice. Also, by the choice of A , at this point, $\ell(ua_i) = 1$. The goal now, is to relabel some of the ua_i 's with 3 in such a way that u is not in conflict with the vertices of A . To show this can be achieved, we use the Combinatorial Nullstellensatz (see [4] or Theorem 1.77).

For every $i \in \{1, \dots, r\}$, let X_i be a variable belonging to $S_i = \{0, 1\}$ and representing whether ua_i is assigned label 3 ($X_i = 1$) or not ($X_i = 0$). Let P be the following polynomial:

$$P(X_1, \dots, X_r) = \prod_{i=1}^r \left(\sum_{\substack{j=1 \\ j \neq i}}^r X_j + N_b - n_i \right).$$

Since $r \geq N_{an} \geq 2$, note that P has degree r . Furthermore, the monomial $\prod_{i=1}^r X_i$ has non-zero coefficient (since every X_i has positive coefficient in the description of P). Hence the Combinatorial Nullstellensatz applies and there is a way to choose values x_1, \dots, x_r in $\{0, 1\}$ for X_1, \dots, X_r so that $P(x_1, \dots, x_r) \neq 0$.

For every $i \in \{1, \dots, r\}$ for which $x_i = 1$, set $\ell(ua_i) = 3$. Note that $d_3(u) = N_b + \sum_{j=1}^r x_j$ and $d_2(u) = 0$. Now, if $d_3(v_i) = d_3(u)$ (in which case this value is 1) for some bad component C_i , then set $\ell(v_i w_i) = 3$. This makes u and v_i have different 3-degrees, and similarly for v_i and w_i .

Suppose that there is a conflict in H between two vertices $x \in V_1$ and $y \in V_2$. For now, suppose that u is not one of these two vertices. If $d_2(x) > 0$, then x and y are 2-monochromatic but we did not modify their 2-degrees in this case, and, hence, $d_2(x)$ is even while $d_2(y)$ odd, a contradiction. If x and y are 1-monochromatic, then y must be some w_i in a bad component C_i . By definition of a bad component, w_i can only be in conflict with v_i , which is now 3-monochromatic, a contradiction. Special vertices only belong to V_2 (in nice components), and, hence, they cannot be involved in conflicts. If x and y are 3-monochromatic, then, again, the only vertex of V_2 which might be 3-monochromatic is some w_i in a bad component C_i . Thus, for $y = w_i$, we have $d_3(w_i) = 1$, and hence x cannot be v_i (since $d_3(v_i) = 2$) and must instead be some other neighbor z of u which belongs to A . In particular, z was 1-monochromatic at the beginning of this case, and, thus, w_i and z were already in conflict before, meaning that C_i was a tricky component and not a bad component, which is impossible.

Hence, every possible conflict must involve u . Vertex u has three types of neighbors: those with non-zero 2-degree, the vertices of A , and the v_i 's with $i \leq q$ such that C_i is a bad component. Since $d_2(u) = 0$, the first group of neighbors cannot be in conflict with u . The third group of neighbors cannot be in conflict with u since we made

sure that they do not have the same 3-degree as u through the application of the Combinatorial Nullstellensatz. This leaves us with the second group of neighbors, the vertices of A . Suppose that $a_i \in A$ is in conflict with u . Note that $d_3(a_i) = n_i + x_i$. Since $d_3(a_i) = d_3(u) = N_b + \sum_{j=1}^r x_j$, we have $\sum_{j=1, j \neq i}^r x_j + N_b - n_i = 0$ and thus $P(x_1, \dots, x_r) = 0$, a contradiction.

Hence there is no conflict in H , and H verifies Property (\mathcal{P}_3) . □

We are now ready to get rid of the last possible conflicts in \mathcal{H} .

Claim 10.15. *For every remaining H , we can relabel edges so that H verifies Property (\mathcal{P}_3) .*

Proof. Let $v \in V_1$ and $u \in V_2$ be two adjacent 1-monochromatic vertices of H (which must exist as otherwise H would verify Property (\mathcal{P}_3)). Because H has at least two edges (as otherwise it would belong to M , not to \mathcal{H}), at least one of v and u must have another neighbor in H . Since Claim 10.14 does not apply, note that u must have degree 1 in H (since all neighbors of u in H must be 1-monochromatic due to Claim 10.13 not applying). So v is also adjacent to $k \geq 1$ vertices $x_1, \dots, x_k \in V_2$ different from u , which must all be 2-monochromatic (because of incident downward edges with V_3, \dots, V_t ; recall that all edges of H are labelled 1) as otherwise Claim 10.14 would apply.

Set $H' = H - u$. According to Lemma 10.12, we can relabel edges in H' with 1 and 2 so that all vertices in $(V_1 \cap V(H')) \setminus \{v\}$ have odd 2-degree, while all vertices in $V_2 \cap V(H')$ have even 2-degree. Recall that u is 1-monochromatic. Thus, if also v is 2-monochromatic with odd 2-degree, then we are done. Assume thus that v is 2-monochromatic with even 2-degree.

- Assume first that the 2-degree of v is even at least 2. In that case, set $\ell(vu) = 3$. This way, u becomes 3-monochromatic, while v becomes special.
- Assume now v is 1-monochromatic. This implies that $\ell(vx_1) = 1$. Change $\ell(vx_1)$ to 3. This way, x_1 becomes special (recall its 2-degree is even and at least 1, due to incident downward edges), while v becomes 3-monochromatic. Note that u remains 1-monochromatic.

In both cases, it can be checked that H now fulfils Property (\mathcal{P}_3) . □

At this point, we dealt with all connected components of \mathcal{H} , and the resulting labelling ℓ of G is p-proper by Lemma 10.11. The whole proof is thus complete.

10.2 A conjecture for 2-labellings with restricted product conflicts

According to Theorem 10.6, every nice graph G admits a p-proper 3-labelling. It is unfortunately impossible to design p-proper 2-labellings for every nice graph. Indeed consider the path $u_1u_2u_3u_4u_5u_6$. Since u_1 and u_2 must have different products, u_2u_3 must be labelled 2. The same can be said for u_5 and u_6 : the edge u_4u_5 must be labelled 2. Regardless of the label of u_3u_4 , the vertices u_3 and u_4 have the same product. Hence the path P_6 does not admit a p-proper 2-labelling.

We investigate the possibility that every graph admits a 2-labelling ℓ which is “almost” p -proper. By that, we mean that if there are product conflicts by ℓ , then the structures induced by the conflicting vertices are somewhat weak. Formally, for any integer $x \geq 1$, we denote by S_x the set of vertices v of G with $\rho_\ell(v) = x$. Rephrased differently, the product version of the 1-2-3 Conjecture states that every nice graph G admits a 3-labelling such that S_x is an independent set for every $x \geq 1$. We would like to know the smallest class \mathcal{C} of graphs for which every (nice) graph G admits a 2-labelling ℓ such that $G[S_x]$ belongs to \mathcal{C} , for every $x \geq 1$. Note that if \mathcal{C} contains isolated edges, we can drop the “nice” requirement on G .

One particular relaxation of the problem is when $G[S_x]$ is allowed to be some particular kind of forest. One can naturally wonder whether 2-labellings are powerful enough to achieve this goal. As we did not manage to come up with any obvious reason why this could be wrong, we raise the following conjecture.

Conjecture 10.16. *Every graph G can be 2-labelled so that $G[S_x]$ is a forest for every $x \geq 1$.*

It is worth noting that Conjecture 10.16 matches a similar conjecture raised in [82] by Gao, Wang and Wu in the sum context. They notably proved that the sum version of Conjecture 10.16 holds for graphs with maximum average degree at most 3 and series-parallel graphs. Recall that a p -proper 2-labelling is equivalent to an s -proper $\{0, 1\}$ -labelling, this remark still holds for our relaxed setting and translates to a conjecture in the sum setting with labels 0 and 1. In what follows, as support, we prove Conjecture 10.16 (sometimes in an actually stronger form) for three classes of graphs: complete graphs, bipartite graphs, and subcubic graphs.

Theorem 10.17. *Every complete graph K_n admits a 2-labelling such that one of the S_x 's induces an edge, while all other S_x 's are independent sets.*

Proof. We give an iterative labelling scheme which, starting from K_2 , yields a desired 2-labelling for larger and larger complete graphs K_n . To that end, we need a stronger hypothesis, namely that for every complete graph K_n there is a desired 2-labelling with the additional requirement that either, there is no vertex incident only with edges labelled 1, or there is no vertex incident only with edges labelled 2. This implies that there cannot be two vertices u and v in K_n such that u has product 1 and v has product 2^{k-1} . It is nonetheless possible for one of the two products to appear with our labelling.

This is true for K_2 : by assigning label 1 to the only edge, we get a 2-labelling where S_1 induces an edge (while there are no other S_x 's) and no vertex is incident only with edges labelled 2. Assume now our stronger claim is true for K_{n-1} for some $n \geq 3$, and consider a 2-labelling of K_{n-1} , with vertex set $\{v_1, \dots, v_{n-1}\}$, obtained by induction (thus with the desired properties). Let us extend this labelling to the incident edges of a newly-added vertex v_n joined to all vertices in $\{v_1, \dots, v_{n-1}\}$, by assigning label 1 to all edges incident with v_n if no vertex in $\{v_1, \dots, v_{n-1}\}$ is incident only with edges labelled 1, or by assigning label 2 to all edges incident with v_n if no vertex in $\{v_1, \dots, v_{n-1}\}$ is incident only with edges labelled 2. Note that the 2-degree of all vertices in $\{v_1, \dots, v_{n-1}\}$ grows by the same amount, either 0 or 1. Thus, no new conflict involving two vertices in $\{v_1, \dots, v_{n-1}\}$ arises. Now, regarding v_n , its 2-degree is either the smallest possible (0) or the largest possible ($n - 1$) for a vertex with degree $n - 1$. By our choice of making v_n incident with either only edges assigned label 1 or only edges assigned label 2, we deduce that v_n

cannot be involved in a conflict. Thus, there remains only one conflict, and there is either no vertex in $\{v_1, \dots, v_n\}$ incident only with edges labelled 2, or no vertex in $\{v_1, \dots, v_n\}$ incident only with edges labelled 1. This concludes the proof. \square

Theorem 10.18. *Every connected bipartite graph G admits a 2-labelling such that one of the S_x 's induces at most one star and isolated vertices, while all other S_x 's are independent sets.*

Proof. This is a direct consequence of applying Lemma 10.12. Let A and B be the two parts of G . Choose a vertex v in A . Lemma 10.12 ensures that we can relabel the edges of G with labels 1 and 2 so that every vertex of $A \setminus \{v\}$ has odd 2-degree while every vertex of B has even 2-degree. Hence if some conflicts remain, they all involve the vertex v and the corresponding S_x induces a star. \square

Theorem 10.19. *Every subcubic graph G admits a 2-labelling such that all S_x 's induce a forest.*

Proof. We prove the claim by induction on $|V(G)| + |E(G)|$. As the claim can easily be proved when G is small, we focus on proving the general case, which we do by considering the possible cases for the minimum degree $\delta(G)$ of G .

- First assume $\delta(G) = 1$, and let v be a 1-vertex of G . Let us consider $G' = G - v$. By the induction hypothesis, there is a 2-labelling of G' which is as desired. We extend this labelling to G by assigning label 1 to the edge incident with v . This way, note that the resulting labelling is as desired, since $G'[S_1]$ gets added a pending or isolated vertex.
- Next assume $\delta(G) = 2$, and let u be a 2-vertex with neighbors v and w of degree at least 2. We here consider $G' = G - u$, which has a 2-labelling with the desired properties. Let us first try to extend this labelling to G by assigning label 1 to uv and uw . Note that if the desired properties are not met, then it must be because $G'[S_1]$ has a path from v to w . In particular, both v and w have product 1, and each of these two vertices is adjacent, in $G'[S_1]$, to another vertex. Then, assign label 2 to uv and label 1 to uw . Now the resulting labelling of G must be as desired, since this removed v from $G'[S_1]$, and added a pending or isolated path of length 2 to $G'[S_2]$. This is because G is subcubic, which, at this point, implies that v has at most one neighbor in S_2 .
- Lastly assume $\delta(G) = 3$, *i.e.* G is cubic, and consider u a 3-vertex with neighbors v, w, x of degree 3. We consider $G' = G - u$, which, again, has a 2-labelling with the desired properties. If we do not obtain a desired labelling of G when assigning label 1 to uv, uw and ux , then it must be because, say, v and w have product 1 and are joined by a path in $G'[S_1]$. By arguments above, due to the bounded maximum degree of G , if we do not obtain a desired labelling when assigning label 2 to uv and label 1 to uw and ux , then this must be because x has product 2, and $G'[S_2]$ contains a path from x to a neighbor of v . Then we deduce that, by the labelling of G' , in G' the two remaining neighbors of v have product 1 and 2, and x has a neighbor with product 2. Then note that we are done when assigning label 2 to uv and ux , and label 1 to uw . Indeed, this removes v from $G'[S_1]$ and x from $G'[S_2]$, adds to $G'[S_2]$ a pending or isolated edge (attached to v), and adds to $G'[S_4]$ a pending or isolated path of length 2 (attached to u).

This concludes the proof. \square

We proved Conjecture 10.16 for three classes of graphs. The first two results are stronger than the general conjecture. For subcubic graphs, there is actually some leeway in the proof that may allow to have only a few degree 3 vertices in each $G[S_x]$. Unfortunately, it seems quite complicated to use similar arguments in more general contexts making Conjecture 10.16 somewhat difficult to attack.

10.3 On the list variant of p-proper labellings

In this section, we explore the variant of the p-proper k -labellings where the label of each edge e of a graph must belong to some list $L(e)$ of possible labels.

Let L be a k -list edge-assignment, recall that an L -labelling (i.e. an L -edge-coloring) ℓ of G is a labelling where each edge is assigned a label from its list, i.e. $\ell(e) \in L(e)$ for every $e \in E(G)$. Note that the notion of p-proper labellings can be extended naturally to L -labellings. Now, for a nice graph G , we define $\text{ch}_\Pi(G)$ as the smallest k such that G admits a p-proper L -labelling for every k -list assignment L where the labels belong to the set of real numbers.

This definition is inspired by the sum version of this problem. For a nice graph G , the number $\text{ch}_\Sigma(G)$ is the smallest k such that G admits an s-proper L -labelling for every k -list assignment L where the labels belong to the set of real numbers. The List 1-2-3 Conjecture, introduced in 2009 by Bartnicki, Grytczuk and Niwczyk in [14], is the straight analogue of the 1-2-3 Conjecture to the list version.

Conjecture 10.20 (List 1-2-3 Conjecture [14]). *If G is a nice graph, then $\text{ch}_\Sigma(G) \leq 3$.*

The List 1-2-3 Conjecture is of course much stronger than the original conjecture, and, as a matter of fact, there is still no known general constant upper bound on ch_Σ . To date, the best bound we know of, is that $\text{ch}_\Sigma(G) \leq \Delta(G) + 1$ holds for every nice graph G [55]. Constant upper bounds were established for some classes of graphs; see later Section 10.3.2 for more details.

In the rest of this chapter, we study the parameter ch_Π . To the best of our knowledge, this parameter was, to date, only briefly discussed by Seamone in his survey [171], in which he suggests a few of its properties. This parameter is also somewhat close to other studied parameters, such as the notions of *product irregularity strength* [6] (related to labellings for which all vertices, not only the adjacent ones, must be incident with distinct products of labels) and *neighbor-product-distinguishing index* [142] (related to labellings for which the labels assigned to the edges must form a proper edge-coloring, it is a stronger version of AVD coloring).

10.3.1 Early remarks on the parameter ch_Π

As remarked in [171], note that, given an edge uv of a graph G , if $\ell(uv) = 0$ by a labelling ℓ of G , then ℓ cannot be p-proper, since this would imply $\rho_\ell(u) = \rho_\ell(v) = 0$. Thus, for any list assignment L of G , a p-proper L -labelling is actually a p-proper L^* -labelling, where L^* is the list assignment of G verifying $L^*(e) = L(e) \setminus \{0\}$ for every edge $e \in E(G)$. Therefore, throughout this work, we consider list assignments not assigning label 0 to the edge lists. To catch this point, we refine the parameter $\text{ch}_\Pi(G)$ of a graph G to the

parameter $\text{ch}_{\Pi}^*(G)$, which is the smallest $k \geq 1$, if any, such that G admits p-proper L -labellings for every k -list assignment L not assigning label 0.

By the previous remarks, obviously the following holds.

Observation 10.21. *If G is a nice non-empty graph, then $\text{ch}_{\Pi}(G) = \text{ch}_{\Pi}^*(G) + 1$.*

Note that the set in which the labels are taken is extremely important. We already mentioned the role of the label 0 but other subtleties appear when the labels can be negative or even complex numbers. Note that it does not make much sense to limit ourselves to only positive real numbers. As suggested by Seamone in [171], there is a straight connection between the parameters ch_{Σ} and ch_{Π}^* , which follows from the product rule of logarithms (see Section 10.3.2 for more details). If we limit ourselves to only positive real numbers then the two problems become equivalent. Introducing negative numbers makes the problem interesting as his own. If we also allow complex numbers, then we might end up with a finite number of possible products. Indeed let \mathbb{U}_k be the set of all k -roots of 1 in \mathbb{C} . If there is a p -proper \mathbb{U}_k -labelling of a graph G , then every product must belong to \mathbb{U}_k and thus $\chi(G)$ must be at most k . Even though allowing complex numbers would be interesting, we consider from now on that the labels must be non-zero real numbers.

Our goal is to determine possible values for/bounds on $\text{ch}_{\Pi}^*(G)$ for any nice graph G . We note that if L is the 1-list assignment of G where $L(e) = \{1\}$ for every edge e , then G admits no p-proper L -labellings, since every such labelling ℓ would verify $\rho_{\ell}(u) = \rho_{\ell}(v) = 1$ for every edge $uv \in E(G)$.

Observation 10.22. *There is no non-empty graph G verifying $\text{ch}_{\Pi}^*(G) = 1$.*

Analogous conclusions can be reached regarding graphs G with $\text{ch}_{\Pi}^*(G) = 2$. Here, consider the 2-list assignment L of G where $L(e) = \{-1, 1\}$ for every edge e . Then, by an L -labelling ℓ of G , we have $\rho_{\ell}(v) \in \{-1, 1\}$ for every vertex $v \in V(G)$. This implies that ℓ is p-proper if and only if ρ_{ℓ} is a proper 2-coloring of G . This case is a particular case of the labels belonging to \mathbb{U}_k mentioned previously. In turn, this yields the following (also mentioned in [171]):

Observation 10.23. *If G is a graph with $\text{ch}_{\Pi}^*(G) = 2$, then G is bipartite.*

The previous condition is not sufficient, however, as nice connected bipartite graphs G with $\text{ch}_{\Pi}^*(G) = 2$ must fulfil an additional property.

Proposition 10.24. *Let G be a connected bipartite graph with bipartition (A, B) . If $\text{ch}_{\Pi}^*(G) = 2$, then at least one of $|A|$ and $|B|$ must be even.*

Proof. Assume the claim is wrong, and let G be a connected bipartite graph with $\text{ch}_{\Pi}^*(G) = 2$ in which the two parts A and B are of odd size. Consider L , the 2-list assignment of G where $L(e) = \{-1, 1\}$ for every edge $e \in E(G)$. As mentioned earlier, by every L -labelling ℓ of G , we have $\rho_{\ell}(v) \in \{-1, 1\}$ for every vertex $v \in V(G)$. Thus, because G is connected, for such an ℓ to be p-proper we must have, say, $\rho_{\ell}(a) = -1$ for every $a \in A$ and $\rho_{\ell}(b) = 1$ for every $b \in B$. For the first condition to occur, for every $a \in A$ there must be an odd number of incident edges labelled -1 by ℓ . Since $|A|$ is odd, this means that we must have an odd number of edges of G labelled -1 by ℓ . For the second condition to occur, for every $b \in B$ there must be an even number of incident edges labelled -1 by ℓ . For that, we must have an even number of edges of G labelled -1 by ℓ , which is a contradiction. \square

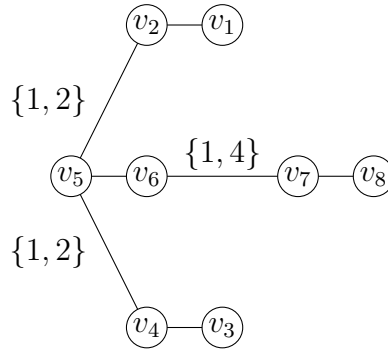


Figure 10.3: A tree with two even part which cannot be labelled with any list assignment L verifying $L(v_5v_2) = \{1, 2\}$, $L(v_5v_4) = \{1, 2\}$ and $L(v_6v_7) = \{1, 4\}$.

Thus, connected graphs G with $\text{ch}_{\Pi}^*(G) = 2$ are connected bipartite graphs with at least one part of even cardinality. This condition is necessary but still not sufficient, however, even in simple graph classes such as trees. To see this, consider the following easy remarks.

Suppose we have a graph G with a pending path wvu of length 2, where $d(u) = 1$ and $d(v) = 2$, and suppose L is a 2-list assignment to the edges of G . Assume more particularly that $L(wv) = \{1, a\}$ for some $a \neq 1$. Then, note that, in any p-proper L -labelling ℓ of G , we cannot have $\ell(vw) = 1$, as otherwise we would have $\rho_{\ell}(v) = \rho_{\ell}(u)$ whatever $\ell(vu)$ is, a contradiction. In other words, the label of wv by a p-proper L -labelling of G is forced to a .

From this, we can construct arbitrarily many trees T with $\text{ch}_{\Pi}^*(T) = 3$ and any wanted cardinality parity for the parts of its bipartition. As an illustration (which admits obvious generalisations), consider the tree T (see Figure 10.3) with vertex set $V(T) = \{v_1, \dots, v_8\}$ and edge set $E(T) = \{v_1v_2, v_2v_5, v_3v_4, v_4v_5, v_5v_6, v_6v_7, v_7v_8\}$, and note that T has no p-proper L -labelling for any list assignment L where $L(v_6v_7) = \{1, a^2\}$ and $L(v_2v_5) = L(v_4v_5) = \{1, a\}$ (for some $a \notin \{1, -1\}$).

10.3.2 Connections with the sum variant, and first bounds on ch_{Π}

As suggested by Seamone in [171], there is a straight connection between the parameters ch_{Σ} and ch_{Π}^* , due to logarithms. Despite this fact being easy to visualize, we give a detailed proof to establish the precise relationship between the two.

Theorem 10.25. *If G is a nice graph, then $\text{ch}_{\Pi}^*(G) \leq 2\text{ch}_{\Sigma}(G) - 1$.*

Proof. Assume we have $\text{ch}_{\Sigma}(G) \leq k$ for some nice graph G and $k \geq 2$. We prove that $\text{ch}_{\Pi}^*(G) \leq 2k - 1$. Let L be a $(2k - 1)$ -list assignment to the edges of G , where none of the $L(e)$'s contains label 0. For every $e \in E(G)$, since $|L(e)| = 2k - 1$, there must be $S(e) \subset L(e)$ such that $|S(e)| = k$ and no two elements of $S(e)$ have the same absolute value. We set $X(e) = \{|x| : x \in S(e)\}$ and $L'(e) = \{\log(x) : x \in X(e)\}$ ¹. Then L' is a k -list assignment of G where each edge e is associated k nonnegative values that are logarithms of values of $L(e)$ with different absolute values.

¹Throughout this chapter, any used log function can be in any fixed base.

Our original assumption $\text{ch}_\Sigma(G) \leq k$ implies that G admits an s-proper L' -labelling ℓ' . We now consider an L -labelling ℓ of G obtained as follows. We consider every edge e of G , and we choose, as $\ell(e)$, any label from $L(e)$ that resulted in $L'(e)$ containing $\ell'(e)$. Thus, ℓ is an L -labelling. As a result, for every $v \in V(G)$ with incident edges e_1, \dots, e_d , we get

$$\sigma_{\ell'}(v) = \sum_{i=1}^d \ell'(e_i) = \sum_{i=1}^d (\log |\ell(e_i)|) = \log \left(\prod_{i=1}^d |\ell(e_i)| \right) = \log(|\rho_\ell(v)|).$$

In particular, ℓ is p-proper since ℓ' is s-proper. □

The connection between ch_Σ and ch_Π^* in Theorem 10.25 implies that, for any constant upper bound on ch_Σ for some graph class, we deduce a constant upper bound on ch_Π^* as well. In the next result, we have listed some constant bounds on ch_Σ from the literature, together with the bounds on ch_Π^* we get as a consequence. It is worth emphasising that we do not claim this list to be exhaustive in any way. Namely, we only list the bounds that seem the most significant to us, and the interested reader has to be aware that more results of the sort can be established from results mentioned in the references below.

Corollary 10.26. *Let G be a nice connected graph.*

- *Since $\text{ch}_\Sigma(G) \leq 5$ (see [213]), we have $\text{ch}_\Pi^*(G) \leq 9$.*
- *Since $\text{ch}_\Sigma(G) \leq \Delta(G) + 1$ (see [55]), we have $\text{ch}_\Pi^*(G) \leq 2\Delta(G) + 1$.*
- *If G is complete, complete bipartite, or a tree, then $\text{ch}_\Sigma(G) \leq 3$ (see [14]); thus $\text{ch}_\Pi^*(G) \leq 5$.*
- *If G is 2-degenerate and non-bipartite, then $\text{ch}_\Sigma(G) \leq 3$ (see [197]); thus $\text{ch}_\Pi^*(G) \leq 5$.*
- *If G is a wheel, then $\text{ch}_\Sigma(G) \leq 3$ (see [159]); thus $\text{ch}_\Pi^*(G) \leq 5$.*
- *If $\text{mad}(G) \leq \frac{11}{4}$, then $\text{ch}_\Sigma(G) \leq 3$ (see [144]); thus $\text{ch}_\Pi^*(G) \leq 5$.*
- *If G is outerplanar, then $\text{ch}_\Sigma(G) \leq 4$ (see [159]); thus $\text{ch}_\Pi^*(G) \leq 7$.*
- *If $\Delta(G) \leq 4$, then $\text{ch}_\Sigma(G) \leq 4$ (see [146]); thus $\text{ch}_\Pi^*(G) \leq 7$.*
- *If G is 2-connected and chordal, or a line graph, then $\text{ch}_\Sigma(G) \leq 5$ (see [196]); thus $\text{ch}_\Pi^*(G) \leq 9$.*

As a consequence of the first item in Corollary 10.26, there is a general constant upper bound on ch_Π^* . In particular, we currently have no evidence that the following, which would be a legitimate guess, might be false.

Conjecture 10.27 (List Multiplicative 1-2-3 Conjecture). *If G is a nice graph, then $\text{ch}_\Pi^*(G) \leq 3$.*

Recall that observations raised at the end of Subsection 10.3.1 establish that this conjecture, if true, would actually be tight.

10.3.3 Algebraic tools

To improve, in next Section 10.4, some of the bounds from Corollary 10.26, we will apply the Combinatorial Nullstellensatz (see Theorem 1.77).

Consider a graph G with edges e_1, \dots, e_m , and a list assignment L to the edges of G . For a vertex u and an edge e of G , we write $e \sim u$ if e is incident with u . Let \vec{G} be any orientation of G . With each edge e_i of G , we associate a variable X_i . Now, we associate with G (through \vec{G}) a polynomial $Q_{\vec{G}}$ with variables X_1, \dots, X_m , being

$$Q_{\vec{G}}(X_1, \dots, X_m) = \prod_{\vec{uv} \in A(\vec{G})} \left(\sum_{e_i \sim u} X_i - \sum_{e_i \sim v} X_i \right).$$

It is easy to see that G has an s -proper L -labelling if and only if there are values $l_1 \in L(e_1), \dots, l_m \in L(e_m)$ such that $Q_{\vec{G}}(l_1, \dots, l_m)$ does not vanish. Bounds on $\text{ch}_{\Sigma}(G)$ can be obtained via the Combinatorial Nullstellensatz through studying the monomials in the expansion of $Q_{\vec{G}}$, more precisely monomials with nonzero coefficient, maximum degree, and, preferably, low exponent values. Note that all the monomials of $Q_{\vec{G}}$ share the very convenient property that they are all of maximum degree m . The tricky part, actually, is about anticipating the coefficients of the monomials of $Q_{\vec{G}}$ (the nonzero ones, particularly), which are far from being obvious in general. In [14], the authors developed a very nice dedicated approach, which is based on studying the permanent of a particular matrix representing $Q_{\vec{G}}$.

A similar polynomial approach can of course be applied for deducing bounds on $\text{ch}_{\Pi}(G)$. The main difference is that, this time, we have to consider the products of labels incident with the vertices, instead of their sums. More precisely, the polynomial of interest is here

$$P_{\vec{G}}(x_1, \dots, x_m) = \prod_{\vec{uv} \in A(\vec{G})} \left(\prod_{e_i \sim u} x_i - \prod_{e_i \sim v} x_i \right).$$

Compared to the polynomial $Q_{\vec{G}}$, a big difference is that, in the expansion of $P_{\vec{G}}$, the monomials are likely to have different degrees, which means that the Combinatorial Nullstellensatz might apply to a few of them only. Even worse is that the degree of $P_{\vec{G}}$ is generally bigger than that of $Q_{\vec{G}}$, and, in particular, the exponents of the monomials generally tend to be bigger too. Note indeed that the degree of $Q_{\vec{G}}$ is precisely m , while the degree of $P_{\vec{G}}$ can be as large as $\sum_{uv \in E(G)} \max\{d(u), d(v)\}$ (which can be reached, e.g. when no two adjacent vertices of G have the same degree).

For these reasons, as will be seen in next Section 10.4, deducing bounds on ch_{Π}^* via the Combinatorial Nullstellensatz only, seems to be viable in particular contexts only.

One particular idea is to translate our product problem into a sum problem. For each variable X_i , let $Y_i = \log(|X_i|)$. Let $P'_{\vec{G}}$ be the following polynomial:

$$P'_{\vec{G}}(Y_1, \dots, Y_m) = \prod_{\vec{uv} \in A(\vec{G})} \left(\sum_{e_i \sim u} Y_i - \sum_{e_i \sim v} Y_i \right).$$

It is easy to see that G has a p -proper L -labelling if there are values $l_1 \in L(e_1), \dots, l_m \in L(e_m)$ such that $P'_{\vec{G}}(\log |l_1|, \dots, \log |l_m|)$ does not vanish. Indeed $P'_{\vec{G}}(\log |l_1|, \dots, \log |l_m|) \neq 0$ implies that for every edge uv :

$$0 \neq \sum_{e_i \sim u} \log |l_i| - \sum_{e_i \sim v} \log |l_i| = \log \left| \prod_{e_i \sim u} l_i \right| - \log \left| \prod_{e_i \sim v} l_i \right|.$$

With this trick, we can transform our product problem into a sum problem. However, some problems arise when doing so. First it is not necessary simple to compute coefficients of the monomials of $P'_{\bar{G}}$, this is why we often apply the Combinatorial Nullstellensatz on just parts of G and apply induction on the rest of the graph. The second problem is even more critical, applying the logarithm can actually reduce the set of possible labels for an edge. Indeed, due to the absolute value, we cannot distinguish between x and $-x$, essentially dividing by 2 our number of labels. Nonetheless, this trick is still viable since not been able to apply it implies that we have many pairs of the form $\{x, -x\}$ in the set of labels for some edges.

10.4 Improved bounds on ch_{Π}^* for some graph classes

We here improve some of the bounds on ch_{Π}^* from Corollary 10.26. We first consider graphs in general, in Subsection 10.4.1. We then focus, in Subsection 10.4.2, on particular classes of graphs, including trees, planar graphs with large girth, and subcubic graphs. In the latter subsection, the exhibited improved bounds are optimal, or close to optimal.

10.4.1 General graphs

The bounds on ch_{Π}^* we establish in this section are expressed as functions of the maximum degree, our goal being to improve the bound of the second item of Corollary 10.25 (which is useful only for graphs with maximum degree at most 6). In order to improve this bound, we must yet again generalize our problem.

A *total list assignment* of G is a function L which assigns to each vertex and to each edge of G , a list of possible labels. For a total L -labelling ℓ , the color $\rho_{\ell}(u)$ of a vertex u is obtained by multiplying the label of u and the labels of the edges incident with u . The notion of p -properness follows naturally. The following result is heavily inspired by a similar result in the sum setting by Ding, Duh, Wang, Wong, Wu, Yu and Zhu [55].

Theorem 10.28. *If G is a nice graph and L is a total list assignment of G such that $|L(uv)| \geq \min(d(u), d(v)) + 2$ for every edge uv and $|L(v)| = 1$ for every vertex v , then G admits a p -proper total L -labelling.*

Proof. The proof of this result is inspired by the proof in [55]. We can suppose that G is connected. Take a vertex u in G with maximum degree, in particular $d(u) \geq 2$ since G is nice (empty graphs are a trivial case). Let v_1, \dots, v_k be the neighbors of u . We create the graph $G' = G - u$ and want to construct a p -proper total L -labelling of G . Note that since there is only one choice for the label of a vertex, we can set $\ell(x)$ to be this label for every vertex x of G .

We classify the neighbors of u according to four types. We say that v_i is a type I neighbor of u , if the connected component of v_i in G' is not an isolated edge. We say that v_i is a type II neighbor of u , if the connected component of v_i in G' is an isolated edge and its only neighbor in G' is some vertex $w_i \notin \{u, v_1, \dots, v_k\}$. If $v_i v_j$ is an isolated edge in G' and $i < j$, then we say that v_i is a type III neighbor of u and that v_j is a type IV neighbor of u . Up to changing the indices of the v_i 's, we can suppose that the other neighbor of a type III neighbor v_i of u is the vertex v_{i+1} . Note that neighbors v_i of u of type II, type III and type IV all have degree 2 and thus $L(uv_i)$ has size at least 4.

For each neighbor v_i of u , we define a_i to be a label from $L(uv_i)$ with minimal absolute value. For every i , let b_i be an element of $L(uv_i) \setminus \{a_i, -a_i\}$ (which exists since $L(uv_i)$ has size at least 3) with minimal absolute value in this set.

In G' , each connected component contains a neighbor of u . Moreover, a connected component is an isolated edge if and only if it does not contain a type I neighbor of u . Let G'' be the graph obtained from G' after removing every isolated edge. Let L' be the labelling of G'' defined by $L'(xy) = L(xy)$ if $xy \in E(G'')$, $L'(x) = L(x)$ if x is not one of the v_i 's and $L'(v_i) = \{\ell(v_i)a_i\}$ for every i . Note that L' verifies the hypotheses of the theorem for the graph G'' . Hence we can apply induction and obtain a total L' -labelling ℓ' of G'' which is p-proper. We obtain a total L -labelling of G by using the labelling ℓ' for the edges on which it is defined. We complete this partial labelling: for every isolated edge xy of G' , we choose $\ell(xy)$ to have maximal absolute value in $L(xy)$ and for every edge uv_i , we choose $\ell(uv_i) = a_i$.

We claim that the total L -labelling ℓ can only have conflicts involving u or conflicts with some type II, III or IV v_i and its only other neighbor. Indeed by construction of L' , every vertex of G'' has the same product in G'' by ℓ' and in G by ℓ .

Suppose that v_i is a type II neighbor of u and is in conflict with w_i , then we set $\ell(uv_i)$ to be the label of $L(uv_i) \setminus \{a_i\}$ with the lowest absolute value. We now have $\rho_\ell(v_i) \neq \rho_\ell(w_i)$. Suppose that v_{i+1} is a type IV neighbor of u and is in conflict with v_i (which is type III), then we set $\ell(uv_{i+1})$ to be the label of $L(uv_{i+1}) \setminus \{a_{i+1}\}$ with the lowest absolute value. We get $\rho_\ell(v_{i+1}) \neq \rho_\ell(v_i)$.

Now the only remaining conflicts are between u and its neighbors. An edge uv_i is safe if $|\rho_\ell(u)| > |\rho_\ell(v_i)|$. In what follows, the absolute value of $\rho_\ell(u)$ never decreases and a safe edge always stays safe.

Suppose that u and v_i are in conflict. Consider the vertex v_{i+1} . There are $d(v_{i+1}) + 2$ labels in $L(uv_{i+1})$. Among those labels, at most $d(v_{i+1}) - 1$ can cause a conflict between v_{i+1} and one of its neighbors (other than u) when labelling uv_{i+1} with them. Hence there are at least three labels for uv_{i+1} which do not create conflicts for v_{i+1} . Let X_{i+1} be this set of labels of uv_{i+1} which do not create conflicts for v_{i+1} . If there is in X_{i+1} a label with absolute value greater than $|\ell(uv_{i+1})|$, then we can relabel uv_{i+1} with this label. This makes uv_i safe (since $|\rho_\ell(u)|$ increases), does not create conflicts for v_{i+1} and keep every previously safe edge, possibly including uv_{i+1} , safe.

If $\ell(uv_{i+1}) = a_{i+1}$ or $\ell(uv_{i+1}) = -a_{i+1}$, then in X_{i+1} , at least one label has absolute value greater than $|a_{i+1}|$ and we can apply the previous procedure. If $|\ell(uv_{i+1})| \neq |a_{i+1}|$, then the label of uv_{i+1} was changed in a previous step. The edge uv_i is not safe since u and v_i are in conflict, hence the label of uv_{i+1} was changed when dealing with conflicts between endpoints of isolated edges of G' . Since $\ell(uv_{i+1}) \neq -a_{i+1}$, we can conclude that $-a_{i+1} \notin L(uv_{i+1})$ and that $\ell(uv_{i+1}) = \pm b_{i+1}$. In particular, since v_{i+1} is a type II or IV neighbor of u , it has degree 2 and $L(uv_{i+1})$ has at least four labels. Hence there is a label c_{i+1} in $L(uv_{i+1}) \setminus \{a_{i+1}, b_{i+1}, -b_{i+1}\}$, which thus has an absolute value greater than $|\ell(uv_{i+1})|$. If our previous remark does not apply then $c_{i+1} \notin X_{i+1}$ and thus, the other neighbor of v_{i+1} is in conflict with v_{i+1} if we label uv_{i+1} with c_{i+1} .

If v_{i+1} is type II, then a_{i+1} is the only color which can create a conflict with w_{i+1} . Indeed, recall that the labels of v_{i+1} , w_{i+1} and uv_{i+1} are the only ones which determine whether v_{i+1} and w_{i+1} are in conflict. Since the two former are fixed and we changed the label of uv_{i+1} from a_{i+1} to its current value, it means that a_{i+1} is the one which creates a conflict between v_{i+1} and w_{i+1} . Hence $c_{i+1} \in X_{i+1}$.

If v_{i+1} is type IV, then v_i is the only other neighbor of v_{i+1} . Choosing a label for $v_i v_{i+1}$ with lower absolute value does not create a conflict between v_i and v_{i+1} , and makes uv_i safe and keep uv_{i+1} safe if it already was.

Since every conflict between u and some v_i makes uv_i safe, these two vertices can never be in conflict again. As we never introduce conflicts not involving the vertex u , there are no more conflicts in G and ℓ is p-proper. \square

By choosing the label of every vertex to be 1, Theorem 10.28 implies the following:

Corollary 10.29. *If G is a nice graph, then $\text{ch}_{\Pi}^*(G) \leq \Delta(G) + 2$.*

10.4.2 Particular classes of graphs

In this subsection, we give tighter results for some classes of graphs.

10.4.2.1 Paths and cycles

Note that Corollary 10.29 implies that four labels for each edge are sufficient when $\Delta(G) \leq 2$, *i.e.* for paths and cycles. In such simple cases, this can actually be refined to a tightest result, proving the List Multiplicative 1-2-3 Conjecture.

Theorem 10.30. *For any integer $n \geq 3$, we have:*

- $\text{ch}_{\Pi}^*(P_n) = 2$ if n is odd or $n = 4$,
- $\text{ch}_{\Pi}^*(P_n) = 3$ otherwise.

For any integer $n \geq 3$, we have:

- $\text{ch}_{\Pi}^*(C_n) = 2$ if $n \equiv 0 \pmod{4}$,
- $\text{ch}_{\Pi}^*(C_n) = 3$ otherwise.

Proof. We deal with cycles first. Let us denote by v_0, \dots, v_{n-1} the successive vertices of C_n and let $e_i = v_i v_{i+1}$ (where, here and further, operations over the indices are understood modulo n). For any two adjacent vertices v_i and v_{i+1} , note that, in order to get $\rho_{\ell}(v_i) \neq \rho_{\ell}(v_{i+1})$ by a labelling ℓ of C_n , we must have $\ell(v_{i-1}v_i) \neq \ell(v_{i+1}v_{i+2})$. Thus, for ℓ to be p-proper, ℓ must be an injective edge-coloring of G .

Now consider G , the graph constructed from C_n by adding one vertex v_{e_i} in G for every edge e_i of C_n . For every i , we add an edge between v_{e_i} and $v_{e_{i+2}}$. The graph G is the graph of conflicts between edges of C_n for an injective edge-coloring. By the remark above, we have $\text{ch}_{\Pi}^*(C_n) = \text{ch}(G)$. Note that G is an odd-length cycle when n is odd, an union of two odd-length cycles when $n \equiv 2 \pmod{4}$, and an union of two even-length cycles when $n \equiv 0 \pmod{4}$. Since even-length cycles have choice number 2 and odd-length cycles have choice number 3 (see e.g. [66]), the result follows.

Regarding paths, remark first that $\text{ch}_{\Pi}^*(P_n) \leq 3$. Indeed for a path $P_n = v_0 v_1 \dots v_{n-1}$, first remove the label 1 from the all lists and label $v_0 v_1$ and $v_1 v_2$ arbitrarily. For $2 \leq i < n - 1$, label $v_i v_{i+1}$ with a label different from the label of $v_{i-2} v_{i-1}$. Remark that no two vertices are in conflict with such a labelling.

For a given $n \geq 3$, similarly as in the case of cycles, let us denote by v_0, \dots, v_{n-1} the successive vertices of P_n , and set $e_i = v_i v_{i+1}$ for every $i \in \{0, \dots, n - 2\}$. Note

that, contrary to the case of cycles, labelling P_n is not similar to coloring the constraint graph G for the injective edge-coloring of P_n , because, when labelling P_n , we must also guarantee that e_1 and e_{n-2} are not assigned label 1, so that v_0 and v_1 are not in conflict, and similarly for v_{n-2} and v_{n-1} .

Let us now consider the following cases.

- If $n \equiv 2 \pmod{4}$, then P_n is a bipartite graph in which the two parts of the bipartition have odd cardinality. As described in Proposition 10.24, we must have $\text{ch}_{\Pi}^*(P_n) > 2$ in such a situation, which implies $\text{ch}_{\Pi}^*(P_n) = 3$.
- If $n = 4$, then first assign to e_1 a label from $L(e_1)$ different from 1, before assigning distinct labels from $L(e_0)$ and $L(e_2)$ to e_0 and e_2 , respectively. Clearly, ℓ is p-proper and $\text{ch}_{\Pi}^*(P_4) = 2$.
- If $n \equiv 0 \pmod{4}$ and $n > 4$, then set $L(e_1) = \{1, 2\}$, $L(e_k) = \{2, 3\}$ for every $k \in \{3, \dots, n-5\}$, and $L(e_{n-3}) = \{1, 3\}$. Since e_1 and e_{n-3} cannot be labelled 1, it must be that $\ell(e_1) = 2$ and $\ell(e_{n-3}) = 3$. We deduce $\ell(e_3) = 3$, $\ell(e_5) = 2$, ..., and $\ell(e_{n-5}) = 3$ since n is a multiple of 4. But then e_{n-5} and e_{n-3} have the same label, making v_{n-4} and v_{n-3} have the same product. Hence $\text{ch}_{\Pi}^*(P_n) > 2$ and thus, $\text{ch}_{\Pi}^*(P_n) = 3$.
- If $n = 3$, then, clearly, we are done when assigning labels from $L(e_0)$ and $L(e_1)$ different from 1 to e_0 and e_1 , respectively.
- If n is odd and $n \geq 5$. We first label the edges e_0, e_2, \dots, e_{n-2} with even index with labels from their respective lists, in such a way that 1) $\ell(e_{n-2}) \neq 1$, and that 2) no two of these edges at distance 2 are assigned the same label. These conditions can clearly be achieved by labelling these edges one by one following the ordering $e_{n-2}, e_{n-4}, \dots, e_2, e_0$. We then achieve the same thing for the edges e_1, e_3, \dots, e_{n-1} with odd index, so that 1) $\ell(e_1) \neq 1$, and that 2) no two of these edges at distance 2 are assigned a same label. Again, this can be easily achieved, e.g. by labelling these edges following the ordering e_1, e_3, \dots, e_{n-1} . By arguments above, ℓ is eventually p-proper. \square

10.4.2.2 Trees

We now prove an upper bound on ch_{Π}^* in the case of trees. The exhibited bound is optimal in general, due to some of the remarks at the end of Subsection 10.3.1. Even some paths attain the upper bound, recall Theorem 10.30.

Theorem 10.31. *If T is a nice tree, then $\text{ch}_{\Pi}^*(T) \leq 3$.*

Proof. The proof is by induction on the number of vertices and edges of T . The base case is when T is a path of length 2, in which situation the claim holds by Theorem 10.30. Thus, we can focus on proving the general case. Let L be a 3-list assignment to the edges of T .

We can assume that T has no pending path of length at least 3, *i.e.* a path uvw such that $d(u) = 1$, $d(v) = d(w) = 2$, and $d(x) \geq 2$. Indeed, assume T has such a path. Let $T' = T - \{u, v\}$. Clearly T' is nice (as otherwise T would be a path, a case for which Theorem 10.30 yields the desired conclusion), and thus T' admits a p-proper L' -labelling ℓ' , where L' denotes the restriction of L to the edges of T' . To extend ℓ' to a p-proper

L -labelling of T , we have to assign to uv and vw labels from their lists, so that no conflict arises. To that aim, we first assign to vw a label different from 1 and from $\frac{\rho_{\ell'}(x)}{\ell'(xw)}$ so that w does not get in conflict with x . Note that this is possible since $|L(vw)| = 3$. Note that, now, because $\ell(vw) \neq 1$, whatever label we assign to uv , we cannot get a conflict between u and v . Thus, when labelling uv , we just need to make sure that v does not get in conflict with w , which can easily be ensured since $|L(uv)| = 3$.

We may also assume that T has branching vertices, *i.e.* vertices with degree at least 3. Indeed, if T has no branching vertex, then T is a path, $\Delta(T) = 2$, and the claim follows from Theorem 10.30. So assume that T has branching vertices. Root T at any branching vertex r . This defines the usual root-to-leaf orientation, through which every non-root vertex has a unique *parent*, *i.e.* a neighbor that is closer to r , and every non-leaf vertex v has *sons*, *i.e.* neighbors that are farther from r , and, more generally, *descendants*, *i.e.* vertices for which the unique path to r goes through v .

Let u be a branching vertex of T that is at farthest distance from r . Note that we have $u = r$ if r is the unique branching vertex of T . By this choice, u has at least two descendants, all of which have degree at most 2. In other words, the descendants of u form $k \geq 2$ disjoint pending paths, none of which has length more than 2, as mentioned earlier. There are then $k = p + q \geq 2$ pending paths attached at u formed by its descendants, where $p \geq 0$ of these paths have length 2, while $q \geq 0$ of them have length 1. We denote by $v_1, \dots, v_p, w_1, \dots, w_q$ the sons of u , where v_1, \dots, v_p belong to pending paths of length 2, while w_1, \dots, w_q are leaves. We also denote by v'_1, \dots, v'_p the neighbor of v_1, \dots, v_p , respectively, different from u . Thus, the v_i 's have degree 2, while the v'_i 's and the w_i 's have degree 1. Lastly, we denote by t the parent of u , if it exists (recall that we have $u = r$ when T has only one branching vertex, in which case u has no parent).

Let $T' = T - \{v_1, \dots, v_p, v'_1, \dots, v'_p, w_1, \dots, w_q\}$. The tree T' is nice, because either r is a branching vertex (case where $u \neq r$) or T' consists in only one vertex (case where $u = r$), and thus T' admits a p -proper L' -labelling ℓ' , where L' denotes the restriction of L to the edges of T' . To extend ℓ' to a p -proper L -labelling of T , we just have to assign labels from their lists to the edges incident with the descendants of u , so that no conflict arises.

We distinguish several cases, based mainly on the value of q .

- Suppose that $q = 0$. Label every edge uv_i with $i \in \{1, \dots, p-1\}$ with an arbitrary label from $L(uv_i)$ different from 1. Now, label uv_p with a label from $L(uv_p)$ different from 1 so that u does not get in conflict with t , if it exists (in case it does not, just assign any label different from 1 to uv_p). Note that this is possible since $|L(uv_p)| = 3$. Lastly, consider every edge $v_i v'_i$. Since $\ell(uv_i) \neq 1$, note that v_i and v'_i cannot get in conflict, whatever label from $L(v_i v'_i)$ is assigned to $v_i v'_i$. Thus, when labelling $v_i v'_i$, we just need to ensure that v_i and u do not get in conflict, which can be avoided since $|L(v_i v'_i)| = 3$.
- Suppose now that $q = 1$. Recall that $p \geq 1$ since $k = p + q \geq 2$. We start by labelling, for every $i \in \{1, \dots, p-1\}$, the edge uv_i with any label different from 1, chosen from $L(uv_i)$. We then consider uv_p , and assign to this edge a label from $L(uv_p)$ different from 1 so that the resulting partial product of u is different from 1. Note that this is possible since $|L(uv_p)| = 3$. Now, note that, by this choice of $\ell(uv_p)$, no matter what $\ell(uw_1)$ is, we cannot get a conflict between u and w_1 . We then assign as $\ell(uw_1)$ a label from $L(uw_1)$ so that u does not get in conflict with t

(if it exists). Lastly, we consider every $i \in \{1, \dots, p\}$, and, to every edge $v_i v'_i$, we assign a value from $L(v_i v'_i)$ so that v_i and u do not get in conflict. This results in ℓ being p -proper. Recall, in particular, that any two v_i, v'_i cannot be in conflict since $\ell(uv_i) \neq 1$.

Suppose now that $q \geq 2$. We start by stating the following general claim:

Claim 10.32. *Let S be a star with center u and $q + 1 \geq 3$ leaves t, w_1, \dots, w_q . Assume we have a partial labelling ℓ' of S where ut is the only edge being assigned a label, a , and that t has (virtual) product $\rho_{\ell'}(t) = A$. If L is a 3-list assignment to the uw_i 's, then, for every $i \in \{1, \dots, q\}$, we can assign a label from $L(uw_i)$ to uw_i , so that ℓ' is extended to a labelling ℓ of S verifying $\rho_{\ell}(u) \notin \{A, \rho_{\ell}(w_1), \dots, \rho_{\ell}(w_q)\}$.*

Proof. Suppose first that $q = 2$. We first assign to uw_1 a label from $L(uw_1)$ different from $1/a$. This way, no matter what label is assigned to uw_2 , note that u and w_2 cannot get in conflict. We now assign a label from $L(uw_2)$ to uw_2 so that the resulting product of u is different from A and the product of w_1 . This is possible since $|L(uw_2)| = 3$.

Assume now that $q \geq 3$. We distinguish the following cases:

- Assume, w.l.o.g., that the three values in $L(uw_1)$ have pairwise distinct absolute values. With each edge uw_i , we associate a variable X_i , and we consider the polynomial

$$P(X_1, \dots, X_q) = \left(a \prod_{i=1}^q X_i - A \right) \cdot \prod_{i=1}^q \left(a \prod_{j=1}^q X_j - X_i \right).$$

For every $i \in \{1, \dots, q\}$, we set $Y_i = \log X_i$. Now considering the polynomial P translates to considering

$$P'(Y_1, \dots, Y_q) = \left(\log(a) + \sum_{i=1}^q Y_i - \log(A) \right) \cdot \prod_{i=1}^q \left(\log(a) + \sum_{j=1}^q Y_j - Y_i \right).$$

Note that, in the expansion of P' , the monomial $Y_1^2 Y_2 \dots Y_q$ has strictly positive coefficient. Thus, by the Combinatorial Nullstellensatz, we can assign values to the Y_i 's so that P' does not vanish, as long as we are given a set of at least three possible distinct values for Y_1 , and a set of at least two possible distinct values for each of Y_2, \dots, Y_q . In turn, this means we can assign values to the X_i 's so that P does not vanish, as long as we have a set of at least three possible values with pairwise distinct absolute values for X_1 , and a set of at least two possible values with distinct absolute values for each of X_2, \dots, X_q . Recall that we made the assumption that the three values in $L(uw_1)$ have pairwise distinct absolute values, while, for every $i \in \{2, \dots, q\}$, there must be at least two values in $L(uw_i)$ with distinct absolute values, since $|L(uw_i)| = 3$. Thus, ℓ' can correctly be extended to ℓ , in the desired way.

- Now assume that every $L(uw_i)$ is of the form $\{\alpha_i, \beta_i, -\beta_i\}$, where α_i and β_i are distinct values with the same sign. Let us start from the labelling ψ of S obtained from ℓ' after setting $\ell(uw_i) = \alpha_i$ for every $i \in \{1, \dots, q\}$. We denote by $s \in \{-, +\}$ the sign of $\rho_{\psi}(u)$, while, for every sign $\epsilon \in \{-, +\}$, we denote by W^{ϵ} the set of vertices w_i for which the sign of $\rho_{\psi}(w_i)$ (thus, of α_i and β_i) is ϵ . Note that W^- and W^+ partition the w_i 's.

To conclude the proof, we consider two last main cases.

- Suppose that $s = +$ and $W^- = \emptyset$. We start by assigning label $-\beta_1$ from $L(uw_1)$ to uw_1 . Note that, as long as each uw_i with $i \in \{2, \dots, q\}$ is assigned a label from $\{\alpha_i, \beta_i\}$, we cannot get a conflict between u and w_i due to their products having different signs. Thus, under that convention, the only conflicts we must pay attention to are along the edges uw_1 and, possibly, ut (in case A is negative).

We here assign a variable X_i to each edge uw_i with $i \in \{2, \dots, q\}$, and consider

$$P(X_2, \dots, X_q) = \left(-\beta_1 a \prod_{i=2}^q X_i - A \right) \cdot \left(-\beta_1 a \prod_{i=2}^q X_i - \beta_1 \right).$$

For every $i \in \{1, \dots, q\}$, we again set $Y_i = \log |X_i|$. Then considering P translates to considering

$$P'(Y_2, \dots, Y_q) = \left(\log(-\beta_1 a) + \sum_{i=2}^q Y_i - \log(A) \right) \cdot \left(\log(-\beta_1 a) + \sum_{i=2}^q Y_i - \log(\beta_1) \right).$$

Recall that $q \geq 3$. Then, whatever q is, in the expansion of P' the monomial $Y_2 Y_3$ has strictly positive coefficient. The Combinatorial Nullstellensatz then implies that we can assign values to Y_2, \dots, Y_q so that P' does not vanish, assuming we have at least two values to choose from for each of Y_2 and Y_3 , and at least one value to choose from for each of Y_4, \dots, Y_q . From this, we deduce that we can assign values to vw_2, \dots, vw_q from $\{\alpha_2, \beta_2\}, \{\alpha_3, \beta_3\}, \{\alpha_4\}, \{\alpha_5\}, \dots, \{\alpha_q\}$, respectively, so that u is in conflict with none of w_1 and t . Recall that the resulting sign of $\rho_\ell(u)$ is negative, while the sign of all vertices w_i with $i \in \{2, \dots, q\}$ is positive. Thus, these vertices also cannot be in conflict.

- Suppose that $s = +$ and $W^- \neq \emptyset$. Assume w.l.o.g. that $w_1 \in W^-$. Recall that, as long as u and w_1 get products with different signs by a labelling, they cannot be in conflict. Thus, we here get our conclusion through the Combinatorial Nullstellensatz, by not modelling the possible conflict between u and w_1 . The precise details are as follows. For every $i \in \{1, \dots, q\}$, let X_i be a variable associated with uw_i . We consider the polynomial

$$P(X_1, \dots, X_q) = \left(a \prod_{i=1}^q X_i - A \right) \cdot \prod_{i=2}^q \left(a \prod_{j=1}^q X_j - X_i \right).$$

For every $i \in \{1, \dots, q\}$, we set $Y_i = \log |X_i|$. Then considering P translates to considering

$$P'(Y_1, \dots, Y_q) = \left(\log(a) + \sum_{i=1}^q Y_i - \log(A) \right) \cdot \prod_{i=2}^q \left(\log(a) + \sum_{j=1}^q Y_j - Y_i \right).$$

In the expansion of P' , the monomial $Y_1 \dots Y_q$ has strictly positive coefficient, and, thus, by the Combinatorial Nullstellensatz, we can assign labels from $\{\alpha_1, \beta_1\}, \dots, \{\alpha_q, \beta_q\}$ to uw_1, \dots, uw_q , respectively, resulting in a labelling ℓ of S where u gets in conflict with none of w_2, \dots, w_q, t . Proceeding that way, recall that the sign of $\rho_\ell(u)$ is positive, while that of $\rho_\ell(w_1)$ is negative. Then, also u and w_1 cannot be in conflict, and ℓ is p-proper.

To conclude the proof, let us point out that the cases where $s = -$ can be treated in a symmetric way, by considering whether W^+ is empty or not. ■

We are now ready to conclude the proof of Theorem 10.31. Recall that we have obtained a labelling ℓ' of $T' = T - \{v_1, \dots, v_p, v'_1, \dots, v'_p, w_1, \dots, w_q\}$ by induction, and that we are in the case where u is adjacent to $q \geq 2$ leaves (and, possibly, p v_i 's and one parent t). We start extending ℓ' to T by considering every edge uv_i (if such edges exist) and assigning to it a label from $L(uv_i)$ different from 1. This is clearly possible, since $|L(uv_i)| = 3$. We now apply Claim 10.32 to the uw_i 's to get all edges incident with u labelled, in such a way that u is not in conflict with any of t (if it exists; if it does not, then note that the claim applies in a very close way) and the w_i 's. The main difference here, is that, though we do not have to care about possible conflict between u and the v_i 's for now, the claim must be employed with taking into consideration the contribution of the uw_i 's to the product of u . In particular, $A = \rho_{\ell'}(t)$ and $a = \ell'(ut) \prod_{i=1}^p \ell'(uv_i)$. Lastly, it remains to label every $v_i v'_i$ with a label from $L(v_i v'_i)$ so that v_i and u do not get into conflict, which is possible since we have three possible labels. Recall in particular that v_i and v'_i cannot be in conflict since $\ell(uv_i) \neq 1$. Eventually, ℓ is p-proper, as desired. □

10.4.2.3 Planar graphs with large girth

Planar graphs with large enough girth are known to be 2-degenerate and to have low maximum average degree. Thus, the fourth and sixth items of Corollary 10.26 establish 5 as a constant upper bound on $\text{ch}_{\Pi}^*(G)$ when G is indeed a nice planar graph with large girth. In what follows, we improve this upper bound down to 4 when $g(G) \geq 16$, getting closer to the List Multiplicative 1-2-3 Conjecture for this class of graphs. Our proof involves arguments that are reminiscent to those used to prove Theorem 10.31, combined together with the following structural result:

Theorem 10.33 (e.g. Nešetřil, Raspaud, Sopena [156]). *If G is a planar graph with girth $g(G) \geq 5\ell + 1$ for some $\ell \geq 1$, then either:*

- $\delta(G) = 1$, or
- G contains an ℓ -thread, i.e. a path $uv_1 \dots v_\ell w$ where $d(u), d(w) \geq 2$, and $d(v_i) = 2$ for every $i \in \{1, \dots, \ell\}$.

We are now ready to prove our result.

Theorem 10.34. *If G is a nice planar graph with girth $g(G) \geq 16$, then $\text{ch}_{\Pi}^*(G) \leq 4$.*

Proof. Assume the claim is wrong, and let G be a minimal counterexample to the claim. We may assume that G is connected, and, due to Theorems 10.30 and 10.31, that $\Delta(G) \geq 3$ and that G is not a tree. Let L be a 4-list assignment to the edges of G . We prove the result by contradicting the existence of G , i.e. by showing that G admits p-proper L -labellings, whatever L is.

If $\delta(G) \geq 2$, then, by Theorem 10.33, we can find a 3-thread $uv_1 v_2 v_3 w$ in G . In that case, we consider $G' = G - v_2$. Note that G' may consist in up to two connected components, each of which has at least two edges (since $d(u), d(w) \geq 2$, by the assumption on $\delta(G)$) and girth at least 16 (in case there is only one connected component, G' might be a tree; in that case, $g(G') = \infty$, and the girth condition remains true). So G' is nice and planar, and, by minimality of G , there is a p-proper L' -labelling ℓ' of G' , where L'

denotes the restriction of L to the edges of G' . To obtain a contradiction, it now suffices to extend ℓ' to a p-proper L -labelling of G , and, for this, we just have to assign labels from $L(v_1v_2)$ and $L(v_2v_3)$ to v_1v_2 and v_2v_3 , respectively, so that no conflict arises. This can clearly be done since $|L(v_1v_2)| = |L(v_2v_3)| = 4$, by first assigning to v_1v_2 a label different from $\ell'(v_3w)$ for which v_1 and u get different partial products, and then assigning to v_2v_3 a label so that v_1 and v_2 are not in conflict, and similarly for v_3 and w .

We may thus assume that $\delta(G) = 1$. Since G is not a tree, this means that, by repeatedly removing vertices of degree 1 while there are some, we end up with a planar connected graph G^- such that $\delta(G^-) \geq 2$ and $g(G^-) \geq 16$. More precisely, for every $v \in V(G) \cap V(G^-)$, we can denote by T_v the pending tree rooted at v in G , which, if $d_G(v) = d_{G^-}(v)$, is reduced to the single vertex v . Then G^- is obtained from G by contracting every T_v to v . For every $v \in V(G) \cap V(G^-)$, we deal, in G , with T_v through the terminology introduced in the proof of Theorem 10.31 (in particular, the notions of parent, son, descendant and branching vertex have the exact same meaning).

Because $g(G^-) \geq 16$, then, by Theorem 10.33, we deduce that G^- has a 3-thread $P = uv_1v_2v_3w$. Note that P also exists back in G , the difference being that v_1, v_2, v_3 might each be the root of a pending tree (denoted $T_{v_1}, T_{v_2}, T_{v_3}$, respectively, following our terminology) that might have edges. In case we have $V(T_{v_i}) = \{v_i\}$ for every $i \in \{1, 2, 3\}$, then note that P is actually a 3-thread in G , in which case a contradiction can be obtained in the similar way as in the previous case $\delta(G) \geq 2$. Thus, in what follows, we assume that some of $T_{v_1}, T_{v_2}, T_{v_3}$ are not reduced to a single vertex.

By arguments similar to some used in the proof of Theorem 10.31, we may assume that none of $T_{v_1}, T_{v_2}, T_{v_3}$ has 1) a non-root branching vertex, or 2) a pending path of length at least 3 (remind, in particular, that in the current context there is even more room for labelling extensions, due to L being a 4-list assignment). This means that each T_{v_i} is a subdivided star with center v_i , where the pending paths attached to v_i (if any) have length 1 or 2.

We start by handling a very particular case, which is when every T_{v_i} has only one edge $v_iv'_i$, *i.e.* is a star with a single edge $v_iv'_i$. In this case, we consider $G' = G - v_2$. A p-proper L' -labelling of G' (where, again, L' denotes the restriction of L to G'), which exists by minimality of G , can then be extended to a p-proper L -labelling of G , a contradiction, by first labelling v_1v_2 with a label from $L(v_1v_2)$ so that no conflict between v_1 and its two neighbors different from v_2 arises, then labelling v_2v_3 with a label from $L(v_2v_3)$ so that 1) no conflict between v_3 and its two neighbors different from v_2 arises, and 2) v_2 gets partial product different from 1; and lastly labelling the edge $v_2v'_2$ of T_{v_2} with a label from $L(v_2v'_2)$ so that no conflict between v_2 and its two neighbors different from v'_2 arises. Recall, in particular, that v_2 and v'_2 cannot be in conflict due to how v_2v_3 was labelled. Note also that lists of four labels are indeed sufficient to achieve this whole process.

In the more general case, let us consider the graph $G' = G - (V(T_{v_1}) \setminus \{v_1\}) - V(T_{v_2}) - (V(T_{v_3}) \setminus \{v_3\})$ (obtained by removing the non-root vertices of T_{v_1} and T_{v_3} , and the whole of T_{v_2}). By arguments used earlier in the case where $\delta(G) \geq 2$, there is a p-proper L' -labelling ℓ' of G' , where L' denotes the restriction of L to the edges of G' . Our goal, to get a final contradiction, is to extend ℓ' in a p-proper way to the edges v_1v_2 , v_2v_3 and those in $T_{v_1}, T_{v_2}, T_{v_3}$, assigning labels from their respective lists, so that a p-proper L -labelling of G results.

We start by assigning labels from $L(v_1v_2)$ and $L(v_2v_3)$ to v_1v_2 and v_2v_3 , respectively, in such a way that, for the resulting partial products of v_1, v_2, v_3 , 1) v_2 is in conflict with

none of v_1 and v_3 , 2) v_1 is not in conflict with u , 3) v_3 is not in conflict with w , and 4) no v_i in $\{v_1, v_2, v_3\}$ for which T_{v_i} contains only one edge, gets product 1 as a result. This is possible to achieve since $|L(v_1v_2)| = |L(v_2v_3)| = 4$. More precisely, this can be achieved by labelling v_1v_2 first and v_2v_3 second if T_{v_1} has only one edge, or by labelling v_2v_3 first and v_1v_2 second otherwise. Recall, in particular, that we have treated separately the case where all of $T_{v_1}, T_{v_2}, T_{v_3}$ have only one edge, so we are not in that case; the fourth condition must thus be fulfilled for at most two of the v_i 's.

It now remains to label the edges from the T_{v_i} 's. We achieve this by considering T_{v_1}, T_{v_2} and T_{v_3} in turn, so that, once every T_{v_i} has been treated, no vertex in $V(T_{v_1}) \cup \dots \cup V(T_{v_i})$ is involved in conflicts, and none of the vertices in $V(T_{v_{i+1}}) \cup \dots \cup V(T_{v_3})$ had its product altered. This way, the desired p -proper L -labelling of G will result once T_{v_3} has been treated. In what follows, we focus on T_{v_1} , but the arguments apply similarly for T_{v_2} and T_{v_3} .

Recall that T_{v_1} consists of some (possibly none) pending paths of length 1 or 2 attached to v_1 . Let us assume that $p \geq 0$ of these paths have length 2, while $q \geq 0$ of them have length 1. We denote by b_1, \dots, b_p the sons of v_1 that belong to the pending paths of length 2, while we denote by c_1, \dots, c_q those from the pending paths of length 1. Finally, for every $i \in \{1, \dots, p\}$, we denote by b'_i the son of b_i in T_{v_1} . By how v_1v_2 was labelled earlier, note that we already have the desired conclusion around v_1 if $p = q = 0$. We thus focus on the cases where $p + q > 0$.

- The cases where $q \in \{0, 1\}$ can be treated quite similarly as the cases $q = 0$ and $q = 1$ in the proof of Theorem 10.31. Namely, we first label the edges $v_1b_1, \dots, v_1b_{p-1}$ (if such edges exist) with labels different from 1 from their respective lists. If $q = 0$, then we label v_1b_p with a label different from 1 from its list, with making sure that the resulting product of v_1 is different from that of u and v_2 . Otherwise, if $q = 1$, then we label v_1b_p with a label different from 1 from its list, with making sure that the resulting partial product of v_1 does not get equal to 1 (if $p = 0$, then recall that this property is already verified at v_1 , due to how v_1v_2 and v_2v_3 have been labelled). Still in the case where $q = 1$, this guarantees that v_1 and c_1 cannot get in conflict no matter how v_1c_1 is labelled; thus, we can label v_1c_1 with a label from its list so that v_1 does not in conflict with u and v_2 . Note that lists of size 4 are sufficient to achieve these conditions in all cases. We lastly label every edge $b_ib'_i$ (if any) with a label from its list, with making sure that b_i does not get in conflict with v_1 . Because v_1b_i was assigned a label different from 1, recall that b_i and b'_i cannot be in conflict.
- The cases where $q = 2$ can be treated quite similarly. Start by labelling every edge v_1b_i (if there are any) with a label different from 1 from its list. Then, label v_1c_1 with a label from its list, so that the resulting partial product of v_1 does not get equal to 1. Last, label v_1c_2 with a label from its list, so that v_1 gets in conflict with none of u, v_2 and c_1 . Note that this is possible, since we do not have to care about a possible conflict between v_1 and c_2 , and $|L(v_1c_2)| = 4$. To conclude, we can eventually label the $b_ib'_i$'s just as in the previous case.

The general case is when $q \geq 3$. We need a generalisation of Claim 10.32 to the current context.

Claim 10.35. *Let S be a star with center u and $q + 2 \geq 5$ leaves t, t', w_1, \dots, w_q . Assume we have a partial labelling ℓ' of S where ut and ut' are the only edges being assigned*

a label, a and a' , respectively, and that t and t' have (virtual) product $\rho_{\ell'}(t) = A$ and $\rho_{\ell'}(t') = A'$. If L is a 4-list assignment to the uw_i 's, then, for every $i \in \{1, \dots, q\}$, we can assign a label from $L(uw_i)$ to uw_i , so that ℓ' is extended to a labelling ℓ of S verifying $\rho_{\ell}(u) \notin \{A, A', \rho_{\ell}(w_1), \dots, \rho_{\ell}(w_q)\}$.

Proof. Note that each $L(uw_i)$ contains two, three or four values with pairwise distinct absolute values. We consider several cases based on that fact.

- Assume, w.l.o.g., that the four values in $L(uw_1)$ have pairwise distinct absolute values. With each edge uw_i , we associate a variable X_i , and we consider the polynomial

$$P(X_1, \dots, X_q) = \left(aa' \prod_{i=1}^q X_i - A \right) \cdot \left(aa' \prod_{i=1}^q X_i - A' \right) \cdot \prod_{i=1}^q \left(aa' \prod_{j=1}^q X_j - X_i \right).$$

For every $i \in \{1, \dots, q\}$, we set $Y_i = \log |X_i|$. Once again P gets translated to

$$P'(Y_1, \dots, Y_q) = \left(\log(aa') + \sum_{i=1}^q Y_i - \log(A) \right) \cdot \left(\log(aa') + \sum_{i=1}^q Y_i - \log(A') \right) \\ \cdot \prod_{i=1}^q \left(\log(aa') + \sum_{j=1}^q Y_j - Y_i \right).$$

In the expansion of P' , the monomial $Y_1^3 Y_2 \dots Y_q$ has strictly positive coefficient. Thus, by the Combinatorial Nullstellensatz, we can assign values to the Y_i 's so that P' does not vanish, as long as we are given a set of at least four possible distinct values for Y_1 , and a set of at least two possible distinct values for each of Y_2, \dots, Y_q . Regarding P , this implies we can assign values to the X_i 's so that P does not vanish, assuming we have a set of a least four possible values with pairwise distinct absolute values for X_1 , and a set of at least two possible values with distinct absolute values for each of X_1, \dots, X_q . This is met in the current case, since $L(uw_1)$ is assumed to have four values with pairwise distinct absolute values, and $|L(uw_i)| = 4$ for every $i \in \{2, \dots, q\}$. Thus, ℓ' can be extended to ℓ as desired.

- Assume now that, w.l.o.g., both $L(uw_1)$ and $L(uw_2)$ include three values with pairwise distinct absolute values. Then the same conclusion as in the previous case can be reached from considering the monomial $Y_1^2 Y_2^2 Y_3 \dots Y_q$ in the expansion of P' .
- We can thus assume that none of the two previous cases applies, *i.e.* that, w.l.o.g., $L(uw_1)$ includes two or three values with pairwise distinct absolute values, while $L(uw_2), \dots, L(uw_q)$ include each exactly two values with pairwise distinct absolute values. In other words, we have $L(uw_i) = \{\alpha_i, -\alpha_i, \beta_i, -\beta_i\}$ for every $i \in \{2, \dots, q\}$, for some distinct α_i, β_i , while $L(uw_1) = \{\alpha_1, -\alpha_1, \beta_1, -\beta_1\}$ or $L(uw_1) = \{\alpha_1, -\alpha_1, \beta_1, \gamma_1\}$, for some distinct $\alpha_1, \beta_1, \gamma_1$. To conclude the proof, we consider a few more cases:

- Assume first that A and A' have the same sign $s \in \{-, +\}$. For every $i \in \{1, \dots, q-2\}$, let us assign to uw_i a label with sign s from its list. Then:

- * If s and the sign of the partial product of u are the same, then we assign to uw_{q-1} a label with sign s from its list, chosen so that the partial product of u gets different from 1. Note that this is possible, since $L(uw_{q-1})$ contains

- two values with sign s . This guarantees that u and w_q cannot be in conflict, whatever the label of uw_q is. We then assign to uw_q a label with sign $-s$ from its list, so that all edges are labelled and no conflict remains. In particular, u gets product with sign $-s$, while only w_q has this property.
- * Otherwise, *i.e.* if s and the sign of the partial product of u are different, then we assign to uw_{q-1} and uw_q a label with sign s from their lists. As a result, no conflict remains, since u is the only vertex with product being of sign $-s$.
- Now assume that A and A' have different signs, say A is positive while A' is negative. We here start by assigning, for every $i \in \{1, \dots, q-2\}$, a positive label to uw_i from its list $L(uw_i)$. Now:
- * If currently u has negative product, then we assign to uw_{q-1} and uw_q a positive label from their respective lists, with making sure that the product of u gets different from A' . This is possible since $L(uw_{q-1})$ and $L(uw_q)$ have two positive values each. Since only u and t' have negative product, no conflict remains.
 - * Otherwise, *i.e.* u currently has positive product, then we first assign a positive label to uw_{q-1} from its list, chosen so that the current product of u does not get equal to 1. This is possible, since $L(uw_{q-1})$ contains two positive values. This guarantees that u and w_q cannot get in conflict. We then assign to uw_q a negative label from $L(uw_q)$, chosen so that u gets product different from A' . This is possible since $L(uw_q)$ contains two negative values. Since only A' and the products of u and w_q are negative, no conflict remains.

In all cases, we end up with the desired labelling ℓ , which concludes the proof. ■

We can now conclude the case $q \geq 3$ of the proof of Theorem 10.34, thus proving the whole statement. We start by labelling every edge v_1b_i (if any) with any label different from 1 from its list $L(v_1b_i)$. We now apply Claim 10.35 to get all v_1c_i 's labelled with labels from their lists, so that v_1 is not in conflict with any of u , v_2 and the c_i 's. This can be done by applying Claim 10.35 with v_1 , u and v_2 playing the role of u , t and t' , respectively, $\rho_{\ell'}(u)$ and $\rho_{\ell'}(v_2)$ playing the role of A and A' , respectively, $\ell'(uv_1) \prod_{i=1}^p \ell(ub_i)$ and $\ell'(v_1v_2)$ playing the role of a and a' , respectively, and the c_i 's playing the role of the w_i 's. It remains to label the $b_i b_i'$'s (if any), and, for each such edge $b_i b_i'$, it suffices to assign a label from its list so that b_i and v_1 do not get in conflict. Recall that we do not have to mind about a possible conflict between b_i and b_i' , since $\ell(v_1b_i) \neq 1$. □

10.4.2.4 Subcubic graphs

We now consider subcubic graphs, *i.e.* graphs with maximum degree 3. Note that, at this point, the best upper bound we have on ch_{Π}^* for these graphs is 5, obtained from Corollary 10.29. We get one step closer to the List Multiplicative 1-2-3 Conjecture for this class of graphs, by lowering the upper bound down to 4 in the next result.

Theorem 10.36. *If G is a nice subcubic graph, then $\text{ch}_{\Pi}^*(G) \leq 4$.*

Proof. Assume the claim is wrong, and consider G a minimal counterexample to the claim. Clearly, G is connected. Let L be a 4-list assignment to the edges of G . We prove below

that G admits a p-proper L -labelling whatever L is, a contradiction. To that aim, we first show that G is cubic:

- Assume first that $\delta(G) = 1$, and consider u a 1-vertex of G with unique neighbor v .
 - Assume first that $d(v) = 2$, and let w denote the second neighbor of v . Set $G' = G - \{u, v\}$. We can assume that G' is nice, as otherwise G would be the path of length 3, in which case even $\text{ch}_{\Pi}^*(G) \leq 3$ holds by Theorem 10.30, a contradiction. Then, by minimality of G , there is a p-proper L' -labelling ℓ' of G' , where L' denotes the restriction of L to the edges of G' . We extend ℓ' to a p-proper L -labelling of G , getting a contradiction, by correctly assigning labels to uv and vw from their respective lists. We first label vw , by assigning a label from $L(vw)$ that is different from 1, and so that w does not get in conflict with any of its at most two other neighbors different from v . Note that this is possible since $|L(vw)| = 4$. We can now extend the labelling to uv by assigning a label from $L(uv)$ so that v does not get in conflict with w . Note that by how vw was labelled, u and v cannot get in conflict.
 - Assume now that $d(v) = 3$, and let w_1, w_2 denote the two neighbors of v different from u . Set $G' = G - \{u, v\}$. We can assume that G' is nice, as otherwise either 1) one of the w_i 's is a 2-vertex adjacent to a 1-vertex, or 2) w_1w_2 exists and both w_1 and w_2 have degree 2. In the former case, we fall into the previous case (where $d(v) = 2$) we have handled. In the latter case, G has only four edges and the claim can be checked by hand. So G' is nice, and, by minimality of G , there is a p-proper L' -labelling ℓ' of G' , where L' denotes the restriction of L to the edges of G' . To extend it to one of G , thus getting a contradiction, we proceed as follows. For every $i \in \{1, 2\}$, note that there are at least two values $a_i, b_i \in L(uw_i)$ that can be assigned to vw_i without causing any conflict between w_i and its at most two neighbors different from v . We assign labels to vw_1 and vw_2 from $\{a_1, b_1\}$ and $\{a_2, b_2\}$, respectively, so that the product of these two labels is different from 1. It then suffices to assign to uv a label from $L(uv)$ so that v gets in conflict with none of w_1 and w_2 , which is possible since $|L(uv)| = 4$. Again, u and v cannot be in conflict due to how vw_1 and vw_2 have been labelled.
- Assume now that $\delta(G) = 2$, and consider u a 2-vertex of G with neighbors v_1, v_2 . By the minimum degree assumption, each of v_1 and v_2 has one or two neighbors different from u . We here consider $G' = G - u$. We can assume that G' is nice, as, because $\delta(G) = 2$, otherwise it would mean that v_1v_2 is the only other edge, thus that G is C_3 , the cycle of length 3, in which case $\text{ch}_{\Pi}^*(G) \leq 3$ holds by Theorem 10.30, a contradiction. So G' admits a p-proper L' -labelling ℓ' , where L' is the restriction of L to the edges of G' . We show that this p-proper labelling can be extended to uv_1 and uv_2 by assigning labels from their lists, thereby getting a contradiction.

Let X_1, X_2 be variables associated with uv_1 and uv_2 , respectively. Let us denote by y_1, y_2 the values $\rho_{\ell'}(v_1), \rho_{\ell'}(v_2)$, respectively. Let us now consider the polynomial

$$\begin{aligned}
 P(X_1, X_2) = & (X_1X_2 - X_1Y_1) \cdot (X_1X_2 - X_2Y_2) \\
 & \cdot \prod_{w \in N(v_1) \setminus \{u\}} (X_1y_1 - \rho_{\ell'}(w)) \cdot \prod_{w \in N(v_2) \setminus \{u\}} (X_2y_2 - \rho_{\ell'}(w)).
 \end{aligned}$$

If X_1 and X_2 can be assigned values in $L(uv_1)$ and $L(uv_2)$, respectively, so that P does not vanish, then we get a p-proper L -labelling of G . Since X_1 and X_2 are the only variables of P , it is easy to see that, in the expansion of P , the monomial M with largest degree is either $x_1^4x_2^4$ (when $d(v_1) = d(v_2) = 3$), $x_1^3x_2^4$ (when $d(v_1) = 2$ and $d(v_2) = 3$), $x_1^4x_2^3$ (when $d(v_1) = 3$ and $d(v_2) = 2$) or $x_1^3x_2^3$ (when $d(v_1) = d(v_2) = 2$). In all cases, since M has nonzero coefficient, then, by the Combinatorial Nullstellensatz, desired values for X_1 and X_2 can be chosen from lists of size at least 5, thus from lists of size at least 4 if we are guaranteed that they do not include 0 (due to the first two factors of P). From this, we deduce that a p-proper L -labelling of G can be obtained from ℓ' , a contradiction.

Thus, from now on, G can be assumed to be cubic. Let $C = u_1 \dots u_p u_1$ be a smallest induced cycle of G . For every $i \in \{1, \dots, p\}$, we denote by u'_i the neighbor of u_i which does not belong to C . Let $G' = G - E(C)$. Note that G' is nice, since the u_i 's have degree 1 and are not adjacent in G' , while all other vertices have degree 3. Thus, by minimality of G , there is a p-proper L' -labelling ℓ' of G' , where L' denotes the restriction of L to the edges of G' . Our goal is to extend ℓ' to the edges of C in a p-proper way to an L -labelling of G , thereby getting a final contradiction.

To ease the exposition of the upcoming arguments, let us introduce some notation. For every $i \in \{1, \dots, p\}$, we set $L_i = L(u_i u_{i+1})$, $a'_i = \ell'(u_i u'_i)$ and $A'_i = \frac{\rho_{\ell'}(u'_i)}{a'_i}$ (where, here and further, we set $u_{p+1} = u_1$ and $u_0 = u_p$). For some set X of values and $\lambda \in \mathbb{R}^*$, we define $\lambda X = \{\lambda x : x \in X\}$ and $\frac{\lambda}{X} = \{\frac{\lambda}{x} : x \in X\}$. For two sets X and Y , we define $XY = \{xy : x \in X, y \in Y\}$.

The proof goes by distinguishing several cases depending on some lists by L and on the structure of G . In each considered case, it is implicitly assumed that none of the previous cases applies.

1. There are $i_0 \in \{1, \dots, p\}$ and $\alpha \in L_{i_0-1}$ such that, for all $\alpha' \in L_{i_0}$, we have $\alpha\alpha' \neq A'_{i_0}$.

W.l.o.g., assume that $i_0 = 1$. The assumption implies that u_1 and u'_1 can never be in conflict in an extension of ℓ' assigning label α to $u_p u_1$. Let us thus start by assigning label α to $u_p u_1$. We then consider the other edges $u_{p-1} u_p, u_{p-2} u_{p-1}, \dots, u_1 u_2$ of C one by one, following this exact ordering. For every edge $u_i u_{i+1}$ considered that way, we assign a label from $L(u_i u_{i+1})$ chosen in the following manner:

- If $i \in \{3, \dots, p-1\}$, then we assign to $u_i u_{i+1}$ a label so that u_{i+1} is in conflict with neither u_{i+2} nor u'_{i+1} . Note that this is possible since $|L(u_i u_{i+1})| = 4$. In the case where $i = p-1$, we note that $u_{i+2} = u_1$ is a vertex whose product is not fully determined yet; the conflict between u_p and u_1 will actually be taken care of in a later stage of the extension process.
- If $i = 2$, then we assign to $u_2 u_3$ a label so that u_3 is in conflict with neither u_4 nor u'_3 , and the resulting partial product of u_2 gets different from the partial product of u_1 . This is possible, since $|L(u_2 u_3)| = 4$. In case $p = 3$ and, thus, $u_4 = u_1$, the possible conflict between u_3 and u_1 will be handled during the next step of the process.
- If $i = 1$, then we assign to $u_1 u_2$ a label so that u_2 gets in conflict with neither u_3 nor u'_2 , and u_1 and u_p are not in conflict. Again, this is possible because

$|L(u_1u_2)| = 4$. Recall further that u_1 and u_2 cannot be in conflict due to the choice of the label assigned to u_2u_3 . Also, u_1 and u'_1 cannot be in conflict by the initial assumption on α .

Thus, once the whole process has been carried out, we get an L -labelling of G which is p -proper, a contradiction.

Since Case 1 does not apply, then, throughout what follows, for every $i \in \{1, \dots, p\}$, we have

$$L_{i-1} = \frac{A'_i}{L_i} \text{ and } L_i = \frac{A'_i}{L_{i-1}}. \quad (10.1)$$

2. There are $i_0 \in \{1, \dots, p\}$ and $\alpha \in L_{i_0}$ such that, for all $\alpha' \in L_{i_0+2}$, we have $\alpha a'_{i_0+1} \neq \alpha' a'_{i_0+2}$.

W.l.o.g., assume that $i_0 = 1$. The assumption implies that u_2 and u_3 can never be in conflict in an extension of ℓ' assigning label α to u_1u_2 . Let us thus assign label α to u_1u_2 . We then consider the other edges of C , and label them with labels from their respective lists so that no conflict arises. We consider a special value of p , before considering the general case.

- Assume first that $p = 3$, *i.e.* C is a triangle. We start by assigning a label from $L(u_2u_3)$ to u_2u_3 so that u_2 does not get in conflict with u'_2 , and the partial product of u_3 gets different from the partial product of u_1 . Note that this is possible since $|L(u_2u_3)| = 4$. We then assign a label from $L(u_1u_3)$ to u_1u_3 so that u_1 gets in conflict with neither u'_1 nor u_2 , and u_3 does not get in conflict with u'_3 . Again, such a label exists since $|L(u_1u_3)| = 4$. Recall that u_1 and u_3 cannot be in conflict due to how u_2u_3 was labelled. Also, u_2 and u_3 cannot be in conflict by the assumption on α .
- Otherwise, *i.e.* $p \geq 4$, we start by assigning a label from $L(u_2u_3)$ to u_2u_3 so that u_2 and u'_2 do not get in conflict. We then consider the remaining edges $u_pu_1, u_{p-1}u_p, \dots, u_3u_4$ of C one by one, following this exact ordering. For every edge u_iu_{i+1} considered that way, we assign a label from $L(u_iu_{i+1})$ chosen in the following way:
 - If $i \in \{5, \dots, p\}$, then we assign to u_iu_{i+1} a label chosen so that u_{i+1} gets in conflict with neither u_{i+2} nor u'_{i+1} . This is possible since $|L(u_iu_{i+1})| = 4$.
 - If $i = 4$, then we assign to u_4u_5 a label chosen so that u_5 gets in conflict with neither u_6 nor u'_5 , and the partial product of u_4 does not get equal to the partial product of u_3 . This is possible since $|L(u_4u_5)| = 4$.
 - If $i = 3$, then we assign to u_3u_4 a label so that u_4 gets in conflict with neither u_5 nor u'_4 , and u_3 does not get in conflict with u'_3 . Again, this is possible since $|L(u_3u_4)| = 4$. Recall that u_4 and u_3 cannot be in conflict due to how u_4u_5 has been labelled. Also, u_2 and u_3 cannot be in conflict by the assumption on α .

Thus, in all cases, we get a p -proper L -labelling of G , a contradiction.

Since Case 2 does not apply in what follows, then, for every $i \in \{1, \dots, p\}$, we have

$$L_i = \frac{a'_{i+2}}{a'_{i+1}} L_{i+2}. \quad (10.2)$$

3. G is K_4 , the complete graph on four vertices.

Here, C is a cycle $u_1u_2u_3u_1$ of length 3, and we have $u' = u'_1 = u'_2 = u'_3$. Also, ℓ' assigns labels to the three edges incident with u' , since G' is a star. Note that, as long as we label the edges of C last and handle all conflicts at that point, then, prior to labelling C , we might actually change the labels assigned to u_1u' , u_2u' , u_3u' by ℓ' for other labels from their respective lists.

Note now that, for any choice of label a'_3 from $L(u_3u')$ assigned to u_3u' , Identity (10.2) must apply, *i.e.* we must have $L_1 = \frac{a'_3}{a'_2}L_3$, as otherwise previous Case 2 would apply the very same way. This implies that $|L_3| \geq 5$, a contradiction, by the following arguments. Since $|L(u_3u')| = 4$, there are at least two values $x, y \in L(u_3u')$ with distinct absolute values, say $|x| < |y|$. Start by assigning label x to $L(u_3u')$; because Identity (10.2) applies, we deduce that for every $\alpha \in L_1$ we have $x\alpha \in L_3$. The other way around, we have $L_3 = L'_3 = \{x\alpha : \alpha \in L_1\}$ and $|L'_3| = |L_3| = 4$. Now change the label of u_3u' to y . Because $|x| < |y|$, we deduce that, for an $\alpha \in L_1$ with largest absolute value, $y\alpha \notin L'_3$. This implies that L_3 must contain a fifth value not in L'_3 for Identity (10.2) to apply with y .

4. $p = 3$ and C shares an edge with another triangle.

Assume u_1u_2 belongs to a triangle $u'u_1u_2u'$ different from C , where $u' = u'_1 = u'_2$ is the common neighbor of u_1 and u_2 different from u_3 . Because we are not in Case 3, we have $u'_3 \neq u'$, and u' has a neighbor $w \notin V(C)$. Note that, by ℓ' , there are actually three possible values in $L(u'_2u')$ that can be assigned to u'_2u' without causing u' to be in conflict with w , thus two such values x, y , with, say, $|x| < |y|$. Start by setting $a'_2 = y$. By an application of Identity (10.2) (which applies as otherwise Case 2 would), we deduce that $L_1 = \frac{a'_3}{a'_2}L_3$, which reveals the exact four values in L_3 . Now, just as in previous Case 3, we note that by changing the value of a'_2 to x and applying Identity (10.2) again, we deduce that L_3 must contain a fifth value not among the previous four revealed ones. This is a contradiction.

At this point, note that if we modify the label a'_i assigned to any edge $u_iu'_i$ by ℓ' , then this has no impact on the value A'_{i+1} (and, symmetrically, on A'_{i-1}). Indeed, if modifying a'_i also modified A'_{i+1} , then this would imply that $u_iu'_i$ is incident with u'_{i+1} , thus that $u'_i = u'_{i+1}$. But, in this case, we would deduce that $u_iu_{i+1}u'_i$ is a triangle sharing an edge with C , thereby getting a contradiction to the fact that none of Cases 3 and 4 applies.

By manipulating Identities (10.1) and (10.2), note that we can establish the relationship

$$L_i = \frac{a'_{i+2}A'_{i+2}}{a'_{i+1}A'_{i+1}}L_i = \frac{a'_{i+1}A'_{i+1}}{a'_{i+2}A'_{i+2}}L_i \quad (10.3)$$

between any list L_i and some of the a'_i 's and A'_i 's. For every $i \in \{1, \dots, p\}$, we define $\lambda_i = \frac{A'_{i+1}}{a'_{i+2}A'_{i+2}}$; then, $L_i = a'_{i+1}\lambda_iL_i$ by the above.

5. There are $i \in \{1, \dots, p\}$ and a p -proper L -labelling ℓ of G' matching ℓ' on all edges but possibly $u_{i+1}u'_{i+1}$, and such that $|\ell(u_{i+1}u'_{i+1})\lambda_i| \neq 1$.

The definition of ℓ and the fact previous Cases 3 and 4 do not apply, imply that A'_{i+1} , A'_{i+2} and a'_{i+2} are the same by both ℓ' and ℓ . From Identity 10.3, we deduce

that $L_i = \ell(u_{i+1}u'_{i+1})\lambda_i L_i$, where λ_i is the same by both ℓ' and ℓ . Now consider $x_0 \in L_i$; from what we have just deduced, we now get that

$$\left\{ (\ell(u_{i+1}u'_{i+1})\lambda_i)^j x_0 \right\}_{j \in \mathbb{N}} \subseteq L_i.$$

Because $|\ell(u_{i+1}u'_{i+1})\lambda_i| \neq 1$, we then deduce that the set $\left\{ (\ell(u_{i+1}u'_{i+1})\lambda_i)^j x_0 \right\}_{j \in \mathbb{N}}$ has infinite cardinality and is included in L_i , which has size 4; a contradiction.

Note that, by ℓ' , there are actually at least two values in $L(u_i u'_i)$ that could be assigned to $u_i u'_i$ without breaking p-properness. This is because $|L(u_i u'_i)| = 4$, and, when labelling $u_i u'_i$, we only have to make sure that u'_i gets product different from that of its at most two neighbors different from u_i in G' (in particular, note that we must have $A'_i \neq 1$ by ℓ' so that $\rho_{\ell'}(u_i) \neq \rho_{\ell'}(u'_i)$, and thus we do not have to care about u_i and u'_i getting in conflict when relabelling $u_i u'_i$). Because Case 5 does not apply, this actually implies that there are exactly two such values from every $L(u_i u'_i)$, and that these two values are precisely a_i and $-a_i$.

6. *There exists $i \in \{1, \dots, p\}$ such that $L_i \neq \{\alpha, -\alpha, \beta, -\beta\}$ for some distinct $\alpha, \beta \in \mathbb{R}^*$.*

Let us consider the identity $L_i = a'_{i+1}\lambda_i L_i$ again. Since Case 5 does not apply, we have $|\ell'(u_{i+1}u'_{i+1})\lambda_i| = 1$ for any possible value as $\ell'(u_{i+1}u'_{i+1})$ from $L(u_{i+1}u'_{i+1})$. Since u'_{i+1} has, in G' , two neighbors different from u_{i+1} , there are, in $L(u_{i+1}u'_{i+1})$, two possible values for $u_{i+1}u'_{i+1}$ that make u'_{i+1} being not in conflict with these two neighbors, and these at least two possibilities must include a'_{i+1} and $-a'_{i+1}$. Now, by considering the p-proper L' -labelling of G' obtained from ℓ' by changing the label of $u_{i+1}u'_{i+1}$ to $-a_{i+1}$, the same reasoning process leads us to deduce that $L_i = -a'_{i+1}\lambda_i L_i$. This implies that $L_i = -L_i$, a contradiction.

We are now ready to conclude the proof, by considering a few cases on the length of C . The crucial points to keep in mind from now on, are that L verifies, for every $i \in \{1, \dots, p\}$, that 1) $a'_i, -a'_i \in L(u_i u'_i)$ and, in ℓ' , changing the label of $u_i u'_i$ from a'_i to $-a'_i$ cannot raise a conflict in G' , and that 2) there are nonzero real numbers α_i, β_i such that $L_i = \{\alpha_i, -\alpha_i, \beta_i, -\beta_i\}$.

7. *p is even.*

For every $i \in \{1, \dots, p\}$, we associate a variable X_i with the edge $u_i u_{i+1}$. We consider the polynomial

$$P(X_1, \dots, X_p) = \prod_{i=1}^p (X_{i-1} X_i - A'_i),$$

which translates to considering

$$P'(Y_1, \dots, Y_p) = \prod_{i=1}^p (Y_{i-1} + Y_i - \log(A'_i))$$

where $Y_i = \log |X_i|$ for every $i \in \{1, \dots, p\}$. Note that the monomial $Y_1 \dots Y_p$ has maximum degree and nonzero coefficient in the expansion of P' . Thus, by the

Combinatorial Nullstellensatz, we can assign values to the Y_i 's so that P' does not vanish, assuming we have at least two possible values to choose from for each of the Y_i 's. This implies that we can assign values to the X_i 's so that P does not vanish, assuming we have at least two possible values with distinct absolute values to choose from, for each of the X_i 's. Particularly, since $|L(u_i u_{i+1})| = 4$ for every edge $u_i u_{i+1}$, this implies that ℓ' can be extended to the edges of C , resulting in an L -labelling ℓ of G where $\rho_\ell(u_i)$ and $\rho_\ell(u'_i)$ have distinct absolute values for every $i \in \{1, \dots, p\}$. Now, the only possible remaining conflicts are between the u_i 's. Due to all the assumptions made this far, recall, for every $i \in \{1, \dots, p\}$, that ℓ assigns label a'_i to every edge $u_i u'_i$, that $-a'_i \in L(u_i u'_i)$, and that switching $\ell(u_i u_i)$ from a'_i to $-a'_i$ cannot raise a conflict between u'_i and its neighbors. Thus, to get a p -proper L -labelling of G , we can just consider each of the $u_i u'_i$'s in turn, and for each $u_i u'_i$ of them, switch, if necessary, its label to $-a'_i$ so that u_i gets positive product if i is, say, even, or negative product otherwise.

8. $p = 3$.

Because Cases 3 and 4 do not apply, recall that u'_1, u'_2, u'_3 are pairwise different. We extend ℓ' as follows. We start by assigning any label from $L(u_1 u_2)$ to $u_1 u_2$. Next, we assign to $u_3 u_1$ a label from $L(u_3 u_1)$ so that no conflict between u_1 and u'_1 arises, and the resulting partial products of u_2 and u_3 have different absolute values. Note that this is possible, since L_3 is of the form $\{\alpha, -\alpha, \beta, -\beta\}$. We finally assign to $u_2 u_3$ a label from $L(u_2 u_3)$ so that there is no conflict between u_2 and u'_2 , u_3 and u'_3 , and u_1 and u_3 . Recall that u_2 and u_3 cannot be in conflict due to how $u_3 u_1$ was labelled. Thus, the only potential conflict that can remain is between u_2 and u_1 , and, if it occurs, then we can get rid of it by simply changing the label of $u_2 u'_2$ from a'_2 to $-a'_2$. Recall that this cannot make u'_2 get in conflict with its neighbors different from u_2 , and that u_2 and u'_2 also cannot get in conflict unless they already were before switching the label of $u_2 u'_2$.

9. p is odd at least 5.

We first use the Combinatorial Nullstellensatz similarly as in Case 7, to label the edges of C in such a way that, for certain pairs of vertices, the resulting products have distinct absolute values. More precisely, we want to achieve this for the pairs $\{u_1, u'_1\}, \{u_1, u_2\}, \{u_2, u_3\}, \{u_3, u'_3\}, \{u_4, u'_4\}, \{u_5, u'_5\}, \dots, \{u_{p-2}, u'_{p-2}\}$ and $\{u_p, u'_p\}$. We denote by \mathcal{S} the set of those pairs. In order to show that such an extension exists, for every $i \in \{1, \dots, p\}$ we associate a variable X_i with the edge $u_i u_{i+1}$, and consider the polynomial

$$P(X_1, \dots, X_p) = (X_p X_1 - A'_1) \cdot (X_p a'_1 - X_2 a'_2) \cdot (X_1 a'_2 - X_3 a'_3) \\ \cdot \left(\prod_{i=3}^{p-2} (X_{i-1} X_i - A'_i) \right) \cdot (X_{p-1} X_p - A'_p),$$

which, if $Y_i = \log |X_i|$ for every $i \in \{1, \dots, p\}$, is the same as considering

$$P'(Y_1, \dots, Y_p) = (Y_p + Y_1 - \log(A'_1)) \cdot (Y_p + \log(a'_1) - Y_2 - \log(a'_2)) \\ \cdot (Y_1 + \log(a'_2) - Y_3 - \log(a'_3)) \\ \cdot \left(\prod_{i=3}^{p-2} (Y_{i-1} + Y_i - \log(A'_i)) \right) \cdot (Y_{p-1} + Y_p - \log(A'_p)).$$

It can be checked that, in the expansion of P' , the monomial $Y_1 \dots Y_p$ has maximum degree and nonzero coefficient -2 . Thus, by the Combinatorial Nullstellensatz we deduce that there is a way to label the edges of C with labels from their respective lists, so that the desired conflicts (between the adjacent vertices in the pairs of \mathcal{S}) are avoided. In particular, this is possible because all these lists are of the form $\{\alpha, -\alpha, \beta, -\beta\}$, and, in particular, contain two values with distinct absolute values. The resulting labelling might be not p -proper, and, to turn it into a p -proper one, we will *switch* some edges incident with the vertices in C , and, by that, we mean changing the current label l of an edge to $-l$. More particularly, we will switch edges of the form $u_i u_{i+1}$ and $u_i u'_i$; due to some of the assumptions made this far, recall that for every such edge e with current label l , we do have $-l \in L(e)$.

We start by switching, if necessary, $u_2 u'_2$ and $u_{p-1} u'_{p-1}$ so that the products of u'_2 and u'_{p-1} get positive and negative, respectively. Next, we switch $u_1 u_2$, if necessary, so that the product of u_2 gets negative. Now, we consider the edges $u_3 u_4, u_4 u_5, \dots, u_p u_1$ one by one following this ordering, and, for every such considered edge $u_i u_{i+1}$, we switch it, if necessary, so that the product of u_i gets negative if i is odd, and positive otherwise. Lastly, we switch $u_1 u'_1$, if necessary, so that the product of u_1 gets negative.

We claim that the eventual labelling of G is p -proper, our final contradiction. First recall, as mentioned earlier, that the switching operation guarantees that the resulting labelling is an L -labelling. Its p -properness follows from the following arguments. First, for all the pairs of adjacent vertices in \mathcal{S} , the products are different due to distinct absolute values (preserved under the switching operation). Regarding the two adjacent vertices in the pair $\{u_{p-1}, u'_{p-1}\}$, the products have different signs and are thus different. Now, for every two adjacent vertices in the pairs $\{u_3, u_4\}, \{u_4, u_5\}, \dots, \{u_p, u_1\}$, the products are different due to their signs being different. \square

10.5 Perspectives

In this work, we presented two open problems on p -proper labellings.

We proposed a conjecture (Conjecture 10.16) regarding almost p -proper 2-labellings. We were able to prove this conjecture on particular classes of graphs. Regarding the general case, it seems that the problem is quite complicated: the fact that only two labels are available leave very few breathing room when constructing a labelling. Moreover, one of the two labels is the label 1 whose particular behavior regarding products adds even more in complexity to the problem. It may be possible that changing the two possible labels makes the conjecture easier to prove. However, there is no certainty that the conjecture holds for arbitrary labels. We raise the following question.

Question 10.37. *Given any two labels a and b , does every graph G can be $\{a, b\}$ -labelled so that $G[S_x]$ is a forest for every integer x ?*

We showed that the answer to the previous question is “no” if $a = 1$ and $b = -1$ in Observation 10.23. The answer is also “no” if one of the two labels is 0. As these pairs of labels are the only ones yielding a finite set of possible products, it may be that they are the only cases where such a labelling is not achievable.

We also considered the problem of the List Multiplicative 1-2-3 Conjecture. In particular, we have exhibited a few bounds on the parameter ch_{Π}^* , both for graphs in general and for more specific classes of graphs. While some of these bounds are tight, some others remain a bit distant from what we believe should be optimal.

As a main perspective for further work on the topic, it would be nice to obtain a constant upper bound on ch_{Π}^* for graphs in general. Recall that, due to Theorem 10.25, this could be obtained through establishing a constant upper bound on ch_{Σ} . This apart, it would be interesting to verify the List Multiplicative 1-2-3 Conjecture for more classes of graphs. For instance, it would be interesting to improve any of the upper bounds in Corollary 10.26, some of which we have already improved in Subsection 10.4.2. Notably, it is worth mentioning that the arguments used to prove Theorems 10.34 and 10.36 are tight, and, as a result, it seems that our proofs would be hard to improve to lower the bound of 4. From this, we would be interested in having a proof of the List Multiplicative 1-2-3 Conjecture for planar graphs with girth at least 16 or for subcubic graphs.

Index

- (A, B) -path, 22
- (A, B) -separator, 105
- (A, B) -walk, 22
- (A, π_A) -layer, 144
- (m, n) -mixed, 20
- (s_0, s_n) -path, 22
- (s_0, s_n) -walk, 22
- 1-colors, 180
- 1-edge, 180
- 1-monochromatic, 268
- 2-colors, 180
- 2-edge, 180
- 2-edge-neighborhood of v , 180
- 2-monochromatic, 268
- 3-monochromatic, 268
- G_i -layer, 139
- H -coloring, 34
- L -choosable, 37
- L -coloring, 119
- L -edge-choosable, 37
- L -edge-coloring, 37
- L -labelling, 38
- M -sequence of G , 225
- M -smaller, 225
- M_a -reducible, 225
- P -crossing, 197
- P -reducible, 225
- P -sequence of G , 225
- P -smaller, 225
- S -packing edge-coloring, 177
- γ -conflict, 244
- $\{2, 3\}$ -degree, 268
- bc -cycle, 197
- d -degenerate, 26
- i -degree, 268
- k -choosable graph, 37
- k -coloring of signed graphs, 59
- k -connected component, 23
- k -edge-colored, 20
- k -edge-coloring, 36
- k -labelling, 174
- k -list assignment, 37
- k -list edge-assignment, 37
- k -list-assignment, 119
- k -neighbor, 19
- k -th power, 24
- k -vertex, 19
- k^+ -neighbor, 19
- k^+ -vertex, 19
- k^- -neighbor, 19
- k^- -vertex, 19
- s -decomposition, 144
- s -prime, 144
- s -redundant, 150
- s_i -packing, 177
- t -distance k -edge-coloring, 174
- t -distance chromatic index, 174
- uv -partial edge-coloring, 243
 - almost-safe, 243
 - safe, 243
- (partial) proper edge-coloring, 224
- (partial) safe edge-coloring, 224
- 2-SAT, 41
- 3-EDGE-COLORING, 209
- 3-SAT, 42
- k -EDGE-COLORING, 45, 218
- 3-COLORING, 39
- ANNOTATED ODD CYCLE
 - TRANSVERSAL, 77
- ARRAY SEARCH, 39
- COLORING-SIGNED-GRAPHS, 123
- EDGE BIPARTIZATION, 45
- ODD CYCLE TRANSVERSAL, 45
- PLANAR VERTEX 3-COLORING, 46, 213

- SAT-CNF, 41
- VARIABLE DELETION ALMOST 2-SAT, 45
- VERTEX COVER, 43
- (H, π) -COLORING, 64
- ANNOTATED BIPARTITE BALANCED COLORING, 97
- ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING, 101
- CLAUSE DELETION ALMOST 2-SAT, 45
- CONNECTED ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING, 102
- CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING (v2), 102
- CONNECTED HALF ANNOTATED SIGNED- $H_{r,b}^{2-}$ -COLORING, 102
- DISJOINT VERTEX DELETION SIGNED- $H_{r,b}^{2-}$ -COLORING, 101
- DISJOINT VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING, 97
- EDGE DELETION H -COLORING, 68
- EDGE DELETION SIGNED- (H, π) -COLORING, 68
- GROUP DELETION ALMOST 2-SAT, 85
- INJECTIVE k -EDGE-COLORING, 207
- LIMITED SWITCHINGS (H, π) -COLORING, 68
- MULTICOLORED INDEPENDENT SET, 44, 90
- SIGNED- (H, π) -COLORING, 64
- VARIABLE DELETION ALMOST 2-SAT, 85
- VERTEX DELETION H -COLORING, 68
- VERTEX DELETION SIGNED- (H, π) -COLORING, 68, 95
- VERTEX DELETION SIGNED- $H_{-,-}^{2b}$ -COLORING, 97
- VERTEX MULTICUT, 98
- NP, 39
- NP-complete, 41
- NP-hard, 41
- P, 39
- XP, 44
- absolute signed clique number, 62
- acyclic, 26
- adjacence (of edges), 18
- adjacence (of vertices), 18
- adjacency list, 40
- adjacency matrix, 39
- adjacent layers, 139
- adjacent vertex-distinguishing k -coloring, 221
- algorithm, 38
- exponential, 38
- linear, 38
- polynomial, 38
- all-negative, 54
- all-positive, 54
- alternating, 61
- antibalanced, 55
- arboricity, 173
- arc, 19
- AVD k -coloring, 221
- AVD-chromatic index, 221
- average degree, 26
- balanced, 55
- balanced walk, 55
- biased graphs, 65
- bichromatic, 268
- biconnected graph, 23
- bipartite, 24
- Boolean formula, 41
- satisfiability, 41
- border, 159
- border neighbors, 159
- box choice, 230
- Cartesian product, 138, 140
- child, 29
- choice number, 119
- choosability, 37
- chromatic index, 36
- chromatic number, 31, 59
- chromatic number of signed graphs, 59
- clause, 41
- clique, 21
- clique number, 21
- Color Game, 229
- correct assignment, 230
- correct set, 230
- compatible with uv , 224
- complement, 21

-
- complete bipartite graph, 24
 - complete graph, 21
 - configuration, 258
 - degree, 259
 - density, 259
 - weight, 259
 - conflict, 224
 - conflict index, 226
 - conjunctive normal form (CNF), 41
 - connected
 - k -connected graph, 23
 - connected graph, 23
 - connected component, 23
 - connectivity, 23
 - Constraint Satisfaction Problems, 67
 - coordinate system, 139
 - copy (of an edge) in a product, 139
 - core, 34
 - core of a graph, 34
 - creates a conflict for uv , 224
 - crossing, 27
 - cubic graph, 26
 - cut-vertex, 23
 - cycle, 22
 - cyclomatic number, 121

 - decomposition, 138
 - degree, 18
 - degree of an edge, 25
 - dense, 27
 - directed graphs, 19
 - disconnected, 23
 - distance, 24
 - distance of edges, 25
 - distinguishing labellings, 174
 - double switching graph, 61
 - downward edge, 269
 - drawing of a graph, 27
 - duality property, 74

 - edge cut, 23
 - edge-choosability, 37
 - edge-neighborhood of v , 180
 - empty graph, 18
 - endpoints of an edge, 18
 - equivalence class, 47
 - equivalence relation, 47
 - equivalent, 56

 - even cycle, 22
 - Exponential Time Hypothesis (ETH), 43

 - face, 29
 - adjacency, 29
 - degree, 29
 - incidence, 29
 - length, 29
 - factors, 138, 144
 - field, 49
 - finer decomposition, 139
 - finite duality property, 74
 - Fixed Parameter Tractable (FPT), 43
 - flat border configuration, 158
 - forest, 26
 - forget node, 31
 - frustration index, 58

 - gain graph, 64
 - ghost vertices, 254
 - girth, 22
 - good $(1, 2^8)$ -packing edge-coloring, 182
 - greatest common divisor, 139
 - grid, 30
 - group, 48
 - abelian group, 48
 - commutative group, 48

 - hereditary, 27
 - homomorphism of (n, m) -mixed graphs, 35
 - homomorphism, 33, 58
 - homomorphism of directed/oriented graphs, 35
 - homomorphism of multi-graphs, 35
 - horizontal edge, 157

 - identifiable, 59
 - Identifying, 33
 - important (A, B) -separator, 105
 - incidence relation, 18
 - incident (color), 224
 - incident with, 175
 - indegree, 20
 - independence number, 21
 - independent set, 21
 - induced signed subgraph, 63
 - induced subgraph, 20
 - injective k -edge-coloring, 207

- injective chromatic index, [207](#)
- internal vertex, [22](#)
- introduce node, [31](#)
- irregular, [265](#)
- irregularity strength, [265](#)
- isolated vertex, [18](#)

- join node, [31](#)

- label, [37](#), [174](#)
- labelling, [37](#)
- leaf, [26](#)
- leaf node, [31](#)
- lexicographic order, [47](#)
- line graph, [25](#)
- literal, [41](#)
- loop, [18](#)
- loopless graphs, [18](#)

- m-proper labelling, [266](#)
- maximum average degree, [26](#)
- maximum degree, [26](#)
- minimum degree, [26](#)
- mixed graph, [20](#)
- modification problems, [67](#)
- monoid, [48](#)
 - abelian monoid, [48](#)
 - commutative monoid, [48](#)
- monomial, [49](#)
- multi-edge, [18](#)
- multi-graph, [18](#)
- multiple edges, [18](#)
- multiplicity, [18](#)
- multiset, [265](#)

- negative crossing, [163](#)
- negative edges, [54](#)
- neighborhood of a vertex, [18](#)
- nice graph, [265](#)
- nice graphs, [175](#)
- nice tree decomposition, [30](#)
- node, [29](#)
- null graph, [18](#)
- number of horizontal turns, [157](#)
- number of vertical turns, [157](#)

- observability, [221](#)
- odd cycle, [22](#)
- order, [46](#)

- order of a graph, [18](#)
- oriented k -coloring, [35](#)
- oriented chromatic number, [35](#)
- oriented graph, [20](#)
- outdegree, [20](#)
- outer face, [29](#)

- p-proper, [266](#)
- parameter, [43](#)
- parameterized reduction, [44](#)
- parent, [29](#)
- path, [22](#)
- planar embedding, [27](#)
- planar graph, [27](#)
- plane graph, [27](#)
- polynomial, [49](#)
- polynomial duality property, [74](#)
- polynomial reduction, [41](#)
- positive crossing, [163](#)
- positive edges, [54](#)
- positive instance, [39](#)
- prime, [138](#), [144](#)
- prime decomposition, [138](#)
- problem, [39](#)
 - decision problem, [39](#)
- Proof by induction, [46](#)
- Proof by minimal counter-example, [46](#)
- proper, [20](#), [31](#), [36](#)
- proper L -coloring, [37](#)
- proper L -edge-coloring, [37](#)
- pruned graph, [125](#)
- pushable oriented graph, [64](#)
- Pushing, [64](#)

- quotient graph, [156](#)
- quotient group, [48](#)
- quotient set, [47](#)
- quotient walk, [156](#)

- recoloring index, [227](#)
- reducible, [226](#)
- region, [158](#)
 - boundary, [158](#)
 - color, [160](#)
 - type, [158](#)
- regular, [26](#)
- relative signed clique number, [62](#)
- representation, [157](#)
- ring, [48](#)

-
- abelian ring, 48
 - cancellation property, 48
 - commutative ring, 48
 - root, 29
 - rooted tree, 29
 - s-proper labelling, 265
 - semiring, 48
 - abelian semiring, 48
 - cancellation property, 48
 - commutative semiring, 48
 - sign, 54
 - sign-preserving k -coloring of signed graphs, 60
 - sign-preserving L -coloring, 119
 - sign-preserving choice number, 119
 - sign-preserving chromatic number, 60
 - sign-preserving homomorphism, 60
 - signature, 54
 - signed A -layer, 144
 - signed clique, 62
 - signed graph, 54
 - switch-preserving core, 64
 - signed subgraph, 63
 - simple graph, 18
 - size of a graph, 18
 - sp-clique, 62
 - space complexity
 - space on an input, 38
 - space on any input of given size, 38
 - spanning forest, 26
 - spanning tree, 26
 - sparse, 27
 - split of a vertex, 225
 - square graph, 24
 - strong k -edge-choosable, 181
 - strong L -edge-choosable, 181
 - strong L -edge-coloring, 181
 - strong chromatic index, 174
 - strong edge-choosability, 181
 - strong edge-coloring, 174
 - strong product, 168
 - subcubic graph, 26
 - subdivision, 28
 - subforest, 26
 - subgraph, 20
 - subtree, 26
 - subtree of T rooted at u , 29
 - sum of incident labels for a vertex, 265
 - swapping, 269
 - switch-preserving core, 64
 - switching a set of vertices, 55
 - switching a vertex, 55
 - switching core, 64
 - switching core of a signed graph, 64
 - tensor product, 168
 - time complexity
 - time on an input, 38
 - time on any input of given size, 38
 - total list assignment, 288
 - tree, 26
 - tree decomposition, 30
 - bag, 30
 - width, 30
 - treewidth, 30
 - triangle, 22
 - truth values, 41
 - unbalanced walk, 55
 - underlying graph, 54
 - undirected graph, 18
 - upward edge, 269
 - valid partition, 270
 - variable assignment, 41
 - vertex k -coloring, 31
 - vertex L -coloring, 37
 - vertical edge, 157
 - walk, 22
 - closed walk, 22
 - length, 22
 - order, 22
 - weak, 228
 - weak (vertices), 256
 - semi-weak, 256
 - well-founded order, 46

Bibliography

- [1] L. Addario-Berry, R. E. Aldred, K. Dalal, and B. A. Reed. Vertex colouring edge partitions. *Journal of Combinatorial Theory, Series B*, 94(2):237–244, 2005.
- [2] S. Akbari, H. Bidkhori, and N. Nosrati. r -strong edge colorings of graphs. *Discrete Mathematics*, 306:3005–3010, 2006.
- [3] M. O. Albertson and R. Haas. Parsimonious edge coloring. *Discrete Mathematics*, 148:1–7, 1996.
- [4] N. Alon. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1–2):7–29, 1999.
- [5] L. D. Andersen. The strong chromatic index of a cubic graph is at most 10. *Discrete Mathematics*, 108:231–252, 1992.
- [6] M. Anholcer. Product irregularity strength of graphs. *Discrete Mathematics*, 309(22):6434–6439, 2009.
- [7] T. Antal, P. L. Krapivsky, and S. Redner. Social balance on networks: The dynamics of friendship and enmity. *Physica D: Nonlinear Phenomena*, 224(1):130–136, 2006. Dynamics on Complex Networks and Applications.
- [8] K. Appel and W. Haken. Every planar map is four colorable. part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 09 1977.
- [9] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. part II: Reducibility. *Illinois Journal of Mathematics*, 21(3):491–567, 09 1977.
- [10] F. Aurenhammer, J. Hagauer, and W. Imrich. Cartesian graph factorization at logarithmic cost per edge. *Computational complexity*, 2(4):331–349, Dec 1992.
- [11] M. Axenovich, P. Dörr, J. Rollin, and T. Ueckerdt. Induced and weak induced arboricities. *Discrete Mathematics*, 342(2):511–519, 2019.
- [12] P. Balister, O. Riordan, and R. Schelp. Vertex-distinguishing edge-colorings of graphs. *Journal of Graph Theory*, 42:95–109, 2003.
- [13] P. N. Balister, E. Györi, J. Lehel, and R. H. Schelp. Adjacent vertex distinguishing edge-colorings. *SIAM Journal on Discrete Mathematics*, 21:237–250, 2007.
- [14] T. Bartnicki, J. Grytczuk, and S. Niwczyk. Weight choosability of graphs. *Journal of Graph Theory*, 60(3):242–256, 2009.

- [15] O. Baudon, J. Bensmail, T. Davot, H. Hocquard, J. Przybyło, M. Senhaji, E. Sopena, and M. Woźniak. A general decomposition theory for the 1-2-3 Conjecture and locally irregular decompositions. *Discrete Mathematics and Theoretical Computer Science*, 21(1), Apr. 2019.
- [16] Z. Bawar, R. Brewster, and D. Marcotte. Homomorphism duality in edge-coloured graphs. *Annales des Sciences Mathématiques du Québec*, 29(1):21–34, 2005.
- [17] J. Bensmail, H. Hocquard, D. Lajou, and E. Sopena. Further evidence towards the multiplicative 1-2-3 conjecture, 2020. <https://arxiv.org/abs/2004.09090>.
- [18] J. Bensmail, H. Hocquard, D. Lajou, and E. Sopena. On a list variant of the multiplicative 1-2-3 conjecture, 2021. <https://arxiv.org/abs/2102.08052>.
- [19] J. Bensmail, H. Hocquard, D. Lajou, and E. Sopena. A proof of the multiplicative 1-2-3 conjecture, 2021. <https://arxiv.org/abs/2108.10554>.
- [20] M. Bonamy. *Global discharging methods for coloring problems in graphs*. PhD thesis, Université de Montpellier, 2015.
- [21] M. Bonamy, N. Bousquet, and H. Hocquard. Adjacent vertex-distinguishing edge coloring of graphs. In *The Seventh European Conference on Combinatorics, Graph Theory and Applications*, pages 313–318. Springer, 2013.
- [22] M. Bonamy and J. Przybyło. On the neighbor sum distinguishing index of planar graphs. *Journal of Graph Theory*, 85:669–690, 2017.
- [23] O. V. Borodin, A. N. Glebov, A. Raspaud, and M. R. Salavatipour. Planar graphs without cycles of length from 4 to 7 are 3-colorable. *Journal of Combinatorial Theory, Series B*, 93(2):303–311, 2005.
- [24] O. V. Borodin, A. V. Kostochka, and D. R. Woodall. List edge and list total colorings of multigraphs. *Journal of Combinatorial Theory, Series B*, 71(2):184–204, 1997.
- [25] R. Brewster. The complexity of colouring symmetric relational systems. *Discrete Applied Mathematics*, 49(1):95–105, 1994. Special Volume Viewpoints on Optimization.
- [26] R. C. Brewster. *Vertex colourings of edge-coloured graphs*. PhD thesis, Simon Fraser University, 1993.
- [27] R. C. Brewster, R. Dedić, F. Huard, and J. Queen. The recognition of bound quivers using edge-coloured homomorphisms. *Discrete Mathematics*, 297(1):13–25, 2005.
- [28] R. C. Brewster, F. Foucaud, P. Hell, and R. Naserasr. The complexity of signed graph and edge-coloured graph homomorphisms. *Discrete Mathematics*, 340(2):223–235, 2017.
- [29] R. C. Brewster and M. Siggers. A complexity dichotomy for signed H-colouring. *Discrete Mathematics*, 341(10):2768–2773, 2018.

-
- [30] R. L. Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37(2):194–197, 1941.
- [31] M. Brunetti, M. Cavaleri, and A. Donno. Erratum to the article ‘a lexicographic product for signed graphs’. *The Australasian Journal of Combinatorics*, 75:256–258, 2019.
- [32] Y. Bu and C. Qi. Injective edge coloring of sparse graphs. *Discrete Mathematics, Algorithms and Applications*, 10(02):1850022, 2018.
- [33] A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017.
- [34] A. A. Bulatov and D. Marx. Constraint satisfaction parameterized by solution size. *SIAM Journal on Computing*, 43(2):573–616, 2014.
- [35] J. Bulín. On the complexity of H-coloring for special oriented trees. *European Journal of Combinatorics*, 69(C):54–75, Mar. 2018.
- [36] A. Burriss and R. Schelp. Vertex-distinguishing proper edge-colorings. *Journal of Graph Theory*, 26:73–83, 1997.
- [37] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- [38] L. Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.
- [39] D. M. Cardoso, J. Orestes Cerdeira, J. P. Cruz, and C. Dominic. Injective edge coloring of graphs. *Filomat*, 33:6411–6423, 2019.
- [40] D. Cartwright and F. Harary. Structural balance: a generalization of Heider’s theory. *Psychological Review*, 63(4):277–293, 1956.
- [41] J. Cerný, M. Horňák, and R. Soták. Observability of a graph. *Mathematica Slovaca*, 46(1):21–31, 1996.
- [42] G. Chartrand, M. Jacobon, J. Lehel, O. Oellermann, S. Ruiz, and F. Saba. Irregular networks. *Congressus Numerantium*, 64:197–210, 1988.
- [43] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006.
- [44] J. Chen, I. A. Kanj, and G. Xia. Improved parameterized upper bounds for vertex cover. In *Mathematical Foundations of Computer Science 2006*, pages 238–249, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [45] J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC ’08, page 177–186, New York, NY, USA, 2008. Association for Computing Machinery.

- [46] X. Chen and Z. Li. Adjacent-vertex-distinguishing proper edge colorings of planar bipartite graphs with $\delta = 9, 10$ or 11 . *Information Processing Letters*, 115(2):263–268, 2015.
- [47] X. Chen and S. Liu. Adjacent vertex distinguishing proper edge colorings of bicyclic graphs. *International Journal of Applied Mathematics*, 48(4):1–11, 2018.
- [48] R. Chitnis, L. Egri, and D. Marx. List H-coloring a graph by removing few vertices. *Algorithmica*, 78:110–146, 2017.
- [49] R. Cole and L. Kowalik. New linear-time algorithms for edge-coloring planar graphs. *Algorithmica*, 50(3):351–368, 2008.
- [50] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium*, pages 151–158, New York, 1971. ACM.
- [51] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
- [52] C. Crespelle, P. G. Drange, F. V. Fomin, and P. A. Golovach. A survey of parameterized algorithms and the complexity of edge modification, 2020. <https://arxiv.org/abs/2001.06867>.
- [53] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [54] S. Das, S. Nandi, S. Paul, and S. Sen. Chromatic number of signed graphs with bounded maximum degree, 2016. <https://arxiv.org/abs/1603.09557>.
- [55] L. Ding, G.-H. Duh, G. Wang, T.-L. Wong, J. Wu, X. Yu, and X. Zhu. Graphs are $(1, \Delta+1)$ -choosable. *Discrete Mathematics*, 342(1):279–284, 2019.
- [56] M. H. Dolama and Éric Sopena. On the oriented chromatic number of graphs with given excess. *Discrete Mathematics*, 306(13):1342 – 1350, 2006.
- [57] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141(1):109–131, 1995.
- [58] K. Drira. *Coloration d’arêtes ℓ -distance et clustering : études et algorithmes auto-stabilisants*. PhD thesis, Université Claude Bernard Lyon 1, 2010.
- [59] F. Dross, F. Foucaud, V. Mitsou, P. Ochem, and T. Pierron. Complexity of planar signed graph homomorphisms to cycles. *Discrete Applied Mathematics*, 284:166–178, 2020.
- [60] C. Duffy, F. Jacques, M. Montassier, and A. Pinlou. The chromatic number of 2-edge-colored and signed graphs of bounded maximum degree, 2020. <https://arxiv.org/abs/2009.05439>.
- [61] J. Dybizbański, A. Nenca, and A. Szepietowski. Signed coloring of 2-dimensional grids. *Information Processing Letters*, 156:105918, 2020.

-
- [62] K. Edwards, M. Horňák, and M. Woźniak. On the neighbor-distinguishing index of a graph. *Graphs and Combinatorics*, 22(3):341–350, 2006.
- [63] A. Ehrenfeucht, J. Hage, T. Harju, and G. Rozenberg. Complexity issues in switching of graphs. In *Theory and Application of Graph Transformations*, pages 59–70, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [64] P. Erdős. Problems and results in combinatorial analysis and graph theory. In *Graph Theory and Applications*, volume 38 of *Annals of Discrete Mathematics*, pages 81–92. Elsevier, 1988.
- [65] P. Erdős and R. Rado. Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London Mathematical Society*, s3-2(1):417–439, 1952.
- [66] P. Erdős, A. L. Rubin, and H. Taylor. Choosability in graphs. In *Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, volume 26, pages 125–157, 1979.
- [67] E. Estrada and M. Benzi. Are social networks really balanced?, 2014. <https://arxiv.org/abs/1406.2132>.
- [68] R. J. Faudree, A. Gyárfás, R. H. Schelp, and Z. Tuza. The strong chromatic index of graphs. *Ars Combinatoria*, 29B:205–211, 1990.
- [69] O. Favaron, H. Li, and R. Schelp. Strong edge coloring of graphs. *Discrete Mathematics*, 159:103–109, 1996.
- [70] T. Feder. Product graph representations. *Journal of Graph Theory*, 16(5):467–488, 1992.
- [71] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- [72] J. Feigenbaum, J. Hershberger, and A. A. Schäffer. A polynomial time algorithm for finding the prime factors of cartesian-product graphs. *Discrete Applied Mathematics*, 12(2):123–138, 1985.
- [73] B. Ferdjallah, S. Kerdjoudj, and A. Raspaud. Injective edge-coloring of sparse graphs, 2019. <https://arxiv.org/abs/1907.09838>.
- [74] A. Fernández, T. Leighton, and J. L. López-Presa. Containment properties of product and power graphs. *Discrete Applied Mathematics*, 155(3):300–311, 2007.
- [75] F. Foucaud, H. Hocquard, and D. Lajou. Complexity and algorithms for injective edge-coloring in graphs. *Information Processing Letters*, 170:106121, 2021.
- [76] F. Foucaud, H. Hocquard, D. Lajou, V. Mitsou, and T. Pierron. Parameterized Complexity of Edge-Coloured and Signed Graph Homomorphism Problems. In *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:16, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

- [77] F. Foucaud, H. Hocquard, D. Lajou, V. Mitsou, and T. Pierron. Parameterized complexity of edge-coloured and signed graph homomorphism problems, 2019. <https://arxiv.org/abs/1910.01099>.
- [78] J.-L. Fouquet and J.-L. Jolivet. Strong edge-coloring of graphs and applications to multi- k -gons. *Ars Combinatoria*, 16A:141–150, 1983.
- [79] J.-L. Fouquet and J.-M. Vanherpe. On Parsimonious Edge-Colouring of Graphs with Maximum Degree Three. *Graphs and Combinatorics*, 29(3):475–487, 2013.
- [80] Z. Füredi, P. Horák, C. Pareek, and X. Zhu. Minimal oriented graphs of diameter 2. *Graphs and Combinatorics*, 14(4):345–350, Jan. 1998.
- [81] J. A. Gallian. A dynamic survey of graph labeling. *Electronic Journal of combinatorics*, 1(DynamicSurveys):DS6, 2018.
- [82] Y. Gao, G. Wang, and J. Wu. A relaxed case on 1-2-3 conjecture. *Graphs and Combinatorics*, 32(4):1415–1421, 2016.
- [83] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.
- [84] N. Gastineau and O. Togni. On S -packing edge-colorings of cubic graphs. *Discrete Applied Mathematics*, 259:63–75, 2019.
- [85] K. Germina, S. H. K, and T. Zaslavsky. On products and line graphs of signed graphs, their eigenvalues and energy. *Linear Algebra and its Applications*, 435(10):2432–2450, 2011. Special Issue in Honor of Dragos Cvetkovic.
- [86] W. Goddard, S. M. Hedetniemi, S. T. Hedetniemi, J. M. Harris, and D. F. Rall. Broadcast chromatic numbers of graphs. *Ars Combinatoria*, 86:33–49, 2008.
- [87] W. Goddard and H. Xu. The S -packing chromatic number of a graph. *Discussiones Mathematicae Graph Theory*, 32:795–806, 2012.
- [88] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386–1396, 2006.
- [89] G. Hahn, J. Kratochvíl, J. Širáň, and D. Sotteau. On the injective chromatic number of graphs. *Discrete Mathematics*, 256(1):179–192, 2002.
- [90] P. Hall. On Representatives of Subsets. *Journal of the London Mathematical Society*, 10(1):26–30, 1935.
- [91] S. Hameed and G. A. On composition of signed graphs. *Discussiones Mathematicae Graph Theory*, 2012:507–516, 01 2012.
- [92] F. Harary. On the notion of balance of a signed graph. *The Michigan Mathematical Journal*, 2(2):143–146, 1953.
- [93] F. Harary. On the measurement of structural balance. *Behavioral Science*, 4(4):316–323, 1959.

-
- [94] F. Harary and J. A. Kabell. A simple algorithm to detect balance in signed graphs. *Mathematical Social Sciences*, 1(1):131–136, 1980.
- [95] H. Hatami. $\Delta + 300$ is a bound on the adjacent vertex distinguishing edge chromatic number. *Journal of Combinatorial Theory, Series B*, 95(2):246–256, 2005.
- [96] F. Hausdorff. *Grundzüge der Mengenlehre*. Veit and Company, Leipzig, 1914.
- [97] F. Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946. PMID: 21010780.
- [98] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.
- [99] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1):117–126, 1992.
- [100] H. Hocquard. *Colorations de graphes sous contraintes*. PhD thesis, Université Bordeaux 1, 2011.
- [101] H. Hocquard, D. Lajou, and B. Lužar. Between proper and strong edge-colorings of subcubic graphs. In *Combinatorial Algorithms, IWOCA 2020*, volume 12126 of *Lecture Notes in Computer Science*, pages 355–367. Springer, 2020.
- [102] H. Hocquard, D. Lajou, and B. Luzar. Between proper and strong edge-colorings of subcubic graphs, 2020. <https://arxiv.org/abs/2011.02175>.
- [103] H. Hocquard and M. Montassier. Adjacent vertex-distinguishing edge coloring of graphs with maximum degree Δ . *Journal of Combinatorial Optimization*, 26:152–160, 2013.
- [104] H. Hocquard, P. Ochem, and P. Valicov. Strong edge-colouring and induced matchings. *Information Processing Letters*, 113(19):836–843, 2013.
- [105] H. Hocquard and J. Przybyło. On the neighbor sum distinguishing index of graphs with bounded maximum average degree. *Graphs and Combinatorics*, 33:1459–1471, 2017.
- [106] I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981.
- [107] P. Horák, H. Qing, and W. T. Trotter. Induced matchings in cubic graphs. *Journal of Graph Theory*, 17(2):151–160, 1993.
- [108] M. Horňák, D. Huang, and W. Wang. On neighbor-distinguishing index of planar graphs. *Journal of Graph Theory*, 76(4):262–278, 2014.
- [109] D. Huang, Z. Miao, and W. Wang. Adjacent vertex distinguishing indices of planar graphs without 3-cycles. *Discrete Mathematics*, 338(3):139–148, 2015.
- [110] D. Huang, X. Zhang, W. Wang, and S. Finbow. Adjacent vertex distinguishing edge coloring of planar graphs without 3-cycles. *Discrete Mathematics, Algorithms and Applications*, 12(04):2050035, 2020.

- [111] F. Hüffner, N. Betzler, and R. Niedermeier. Separator-based data reduction for signed graph balancing. *Journal of Combinatorial Optimization*, 20(4):335–360, 2010.
- [112] E. Hurley, R. de Joannis de Verclos, and R. J. Kang. An improved procedure for colouring graphs of bounded local density, 2020. <https://arxiv.org/abs/2007.07874>.
- [113] W. Imrich and S. Klavžar. *Product Graphs, Structure and Recognition*. Wiley–Blackwell, 01 2000.
- [114] W. Imrich, S. Klavžar, and D. F. Rall. Cancellation properties of products of graphs. *Discrete Applied Mathematics*, 155(17):2362–2364, 2007.
- [115] W. Imrich and I. Peterin. Recognizing Cartesian products in linear time. *Discrete Mathematics*, 307(3):472–483, 2007. Algebraic and Topological Methods in Graph Theory.
- [116] W. Imrich and I. Peterin. Cartesian products of directed graphs with loops. *Discrete Mathematics*, 341(5):1336–1343, 2018.
- [117] T. Ito, A. Kato, X. Zhou, and T. Nishizeki. Algorithms for finding distance-edge-colorings of graphs. *Journal of Discrete Algorithms*, 5(2):304–322, 2007. 2004 Symposium on String Processing and Information Retrieval.
- [118] L. Jaffke and B. M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *Algorithms and Complexity*, pages 345–356, Cham, 2017. Springer International Publishing.
- [119] E. Jelínková, O. Suchý, P. Hliněný, and J. Kratochvíl. Parameterized problems related to Seidel’s switching. *Discrete Mathematics and Theoretical Computer Science*, 13(2):19–42, 2011.
- [120] G. Joret and W. Lochet. Progress on the adjacent vertex distinguishing edge coloring conjecture. *SIAM Journal of Discrete Mathematics*, 34(4):2221–2238, 2020.
- [121] M. Kalkowski, M. Karoński, and F. Pfender. Vertex-coloring edge-weightings: towards the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 100(3):347–349, 2010.
- [122] R. Kang and P. Manggala. On distance edge-colourings and matchings. *Electronic Notes in Discrete Mathematics*, 34:301–306, 2009. European Conference on Combinatorics, Graph Theory and Applications (EuroComb 2009).
- [123] R. J. Kang and P. Manggala. Distance edge-colourings and matchings. *Discrete Applied Mathematics*, 160(16):2435–2439, 2012.
- [124] F. Kardoš and J. Narboni. On the 4-color theorem for signed graphs. *European Journal of Combinatorics*, 91:103215, 2021. Colorings and structural graph theory in context (a tribute to Xuding Zhu).
- [125] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, 91(1):151–157, 2004.

-
- [126] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
- [127] K.-i. Kawarabayashi and B. Reed. An (almost) linear time algorithm for odd cycles transversal. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 365–378, USA, 2010. Society for Industrial and Applied Mathematics.
- [128] S. Khanna, M. Sudan, L. Trevisan, and D. P. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001.
- [129] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002. Computing and Combinatorics.
- [130] T. Kloks. *Treewidth, Computations and Approximations*. Springer, 1994.
- [131] M. Kochol. Snarks without Small Cycles. *Journal of Combinatorial Theory, Series B*, 67(1):34–47, 1996.
- [132] D. König. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77:453–465, 1916.
- [133] A. Kostochka, X. Li, W. Ruksasakchai, M. Santana, T. Wang, and G. Yu. Strong chromatic index of subcubic planar multigraphs. *European Journal of Combinatorics*, 51:380–397, 2016.
- [134] A. Kostochka, A. Raspaud, and J. Xu. Injective edge-coloring of graphs with given maximum degree. *European Journal of Combinatorics*, 96:103355, 2021.
- [135] C. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15(1):271–283, 1930.
- [136] D. Lajou. On the achromatic number of signed graphs. *Theoretical Computer Science*, 759:50–60, 2019.
- [137] D. Lajou. On Cartesian products of signed graphs. In *Algorithms and Discrete Applied Mathematics*, pages 219–234, Cham, 2020. Springer International Publishing.
- [138] D. Lajou. On Cartesian products of signed graphs. *Discrete Applied Mathematics*, 2021.
- [139] L. Leonid. Universal search problems. *Problems of Information Transmission*, 9(3):115–116, 1973.
- [140] D. Leven and Z. Galil. NP-completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35–44, 1983.
- [141] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

- [142] T. Li, C. Qu, G. Wang, and X. Yu. Neighbor product distinguishing total colorings. *Journal of Combinatorial Optimization*, 33(1):237–253, 2017.
- [143] Y. Li and L. Chen. Injective edge coloring of generalized Petersen graphs. *AIMS Mathematics*, 6(8):7929–7943, 2021.
- [144] Y.-C. Liang, T.-L. Wong, and X. Zhu. Graphs with maximum average degree less than 114 are $(1, 3)$ -choosable. *Discrete Mathematics*, 341(10):2661–2671, 2018.
- [145] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–71, 2011.
- [146] Y. Lu, C. Li, and Z. K. Miao. Weight choosability of graphs with maximum degree 4. *Acta Mathematica Sinica, English Series*, 36(6):723–732, 2020.
- [147] B. Lužar, E. Mačajová, M. Škoviera, and R. Soták. On the conjecture about strong edge-coloring of subcubic graphs. *Manuscript*, 2020.
- [148] K. Lyngsie. On neighbour sum-distinguishing 0,1-weightings of bipartite graphs. *Discrete Mathematics and Theoretical Computer Science (Online Edition)*, 20(1), 2018.
- [149] F. Martin. Frustration and isoperimetric inequalities for signed graphs. *Discrete Applied Mathematics*, 217:276–285, 2017.
- [150] D. Marx. Parameterized coloring problems on chordal graphs. *Theoretical Computer Science*, 351(3):407–424, 2006.
- [151] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, STOC '11*, page 469–478, New York, NY, USA, 2011. Association for Computing Machinery.
- [152] M. Maydanskiy. The incidence coloring conjecture for graphs of maximum degree 3. *Discrete Mathematics*, 292(1–3):131–141, 2005.
- [153] R. Naserasr, Éric Sopena, and T. Zaslavsky. Homomorphisms of signed graphs: An update. *European Journal of Combinatorics*, 91:103222, 2021. Colorings and structural graph theory in context (a tribute to Xuding Zhu).
- [154] R. Naserasr, E. Rollová, and Éric Sopena. Homomorphisms of signed graphs. *Journal of Graph Theory*, 79(3):178–212, 2015.
- [155] C. S. J. Nash-Williams. Decomposition of finite graphs into forests. *Journal of the London Mathematical Society*, 1(1):12–12, 1964.
- [156] J. Nešetřil, A. Raspaud, and É. Sopena. Colorings and girth of oriented planar graphs. *Discrete Mathematics*, 165:519–530, 1997.
- [157] J. Nešetřil and A. Raspaud. Colored homomorphisms of colored mixed graphs. *Journal of Combinatorial Theory, Series B*, 80(1):147–155, 2000.

-
- [158] K. Okrasa and P. Rzażewski. Subexponential algorithms for variants of the homomorphism problem in string graphs. *Journal of Computer and System Sciences*, 109:126–144, 2020.
- [159] H. Pan and D. Yang. On total weight choosability of graphs. *Journal of Combinatorial Optimization*, 25(4):766–783, 2013.
- [160] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [161] C. Payan. *Sur quelques problèmes de couverture et de couplage en combinatoire*. PhD thesis, Institut National Polytechnique de Grenoble - INPG, Université Joseph-Fourier - Grenoble I, 1977. In French.
- [162] K. Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003. Parameterized Computation and Complexity 2003.
- [163] M. Pilipczuk, M. Pilipczuk, and M. Wrochna. Edge bipartization faster than 2^k . *Algorithmica*, 81(3):917–966, 2019.
- [164] J. Przybyło. The 1–2–3 conjecture almost holds for regular graphs. *Journal of Combinatorial Theory, Series B*, 147:183–200, 2021.
- [165] I. Razgon and B. O’Sullivan. Almost 2-SAT is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.
- [166] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32:299–301, 2004.
- [167] N. Robertson and P. D. Seymour. Graph minors. V. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986.
- [168] G. Sabidussi. Graphs with given group and given graph theoretical properties. *Canadian Journal of Mathematics*, 9:515–525, 01 1957.
- [169] G. Sabidussi. Graph multiplication. *Mathematische Zeitschrift*, 72:446–457, 1959/60.
- [170] D. P. Sanders and Y. Zhao. Planar graphs of maximum degree seven are class I. *Journal of Combinatorial Theory, Series B*, 83(2):201–212, 2001.
- [171] B. Seamone. The 1-2-3 conjecture and related problems: a survey, 2012. <https://arxiv.org/abs/1211.5122>.
- [172] P. D. Seymour. On Tutte’s Extension of the Four-Color Problem. *Journal of Combinatorial Theory, Series B*, 31:82–94, 1981.
- [173] M. Sipser. *Introduction to the theory of computation*. PWS Pub. Co, 1st edition, 1997.
- [174] J. Skowronek-Kaziów. Multiplicative vertex-colouring weightings of graphs. *Information Processing Letters*, 112(5):191–194, 2012.

- [175] E. Sopena. The oriented coloring page. <https://www.labri.fr/perso/sopena/pmwiki/index.php?n=TheOrientedColoringPage.TheOrientedColoringPage>.
- [176] R. Soták. Private communication.
- [177] A. Steger and M.-L. Yu. On induced matchings. *Discrete Mathematics*, 120:291–295, 1993.
- [178] S. Strogatz. The enemy of my enemy. *The New York Times*, February 14th 2010.
- [179] P. G. Tait. On the colouring of maps. *Proceedings of the Royal Society of Edinburgh Section A*, 10:501–503, 729, 1880.
- [180] Y. Takenaga and K. Higashide. Vertex coloring of comparability+ke and -ke graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 102–112, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [181] S.-L. Tian and P. Chen. Distance coloring and distance edge-coloring of d- dimensional lattice. In *Intelligent Computing Theories and Applications*, pages 561–568, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [182] G. Toulouse. Theory of the frustration effect in spin glasses: I. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, 9:99, 1987.
- [183] V. G. Vizing. The cartesian product of graphs (In Russian). *Vycisl. Sistemy*, 9:30–43, 1963.
- [184] V. G. Vizing. On an estimate of the chromatic class of a p -graph. *Metody Diskret. Analiz*, 3:25–30, 1964.
- [185] V. G. Vizing. Coloring the vertices of a graph with prescribed colors. *Diskret. Analiz.*, 29:3–10, 1976.
- [186] B. Vučković. Multi-set neighbor distinguishing 3-edge coloring. *Discrete Mathematics*, 341(3):820–824, 2018.
- [187] B. Vučković. Edge-partitions of graphs and their neighbor-distinguishing index. *Discrete Mathematics*, 340(12):3092–3096, 2017.
- [188] W. Wang and Y. Wang. Adjacent vertex distinguishing edge-colorings of graphs with smaller maximum average degree. *Journal of Combinatorial Optimization*, 19:471–485, 2010.
- [189] Y. Wang, J. Cheng, R. Luo, and G. Mulley. Adjacent vertex-distinguishing edge coloring of 2-degenerate graphs. *Journal of Combinatorial Optimization*, 31:874–880, 2016.
- [190] Y. Wang, W. Wang, and J. Huo. Some bounds on the neighbor-distinguishing index of graphs. *Discrete Mathematics*, 338(11):2006–2013, 2015.
- [191] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, September 2000.

-
- [192] A. N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1925–1927.
- [193] Wikipedia, The Free Encyclopedia. Ising model. <http://en.wikipedia.org/w/index.php?title=Ising%20model&oldid=1015490779>, 2021.
- [194] S. G. Williamson. Depth-first search and Kuratowski subgraphs. *Journal of the Association for Computing Machinery*, 31(4):681–693, Sept. 1984.
- [195] P. Winkler. Factoring a graph in polynomial time. *European Journal of Combinatorics*, 8(2):209–212, 1987.
- [196] T.-L. Wong. 2-connected chordal graphs and line graphs are (1, 5)-choosable. *European Journal of Combinatorics*, 91:103227, 2021.
- [197] T.-L. Wong and X. Zhu. Total weight choosability of d-degenerate graphs, 2015. <https://arxiv.org/abs/1510.00809>.
- [198] D. R. Woodall. The average degree of an edge-chromatic critical graph II. *Journal of Graph Theory*, 56(3):194–218, 2007.
- [199] D. R. Woodall. The average degree of a multigraph critical with respect to edge or total choosability. *Discrete Mathematics*, 310:1167–1171, 2010.
- [200] M. Woźniak and M. Horňák. On neighbour-distinguishing colourings from lists. *Discrete Mathematics & Theoretical Computer Science*, 14, 2012.
- [201] J. Wu and W. Lin. The strong chromatic index of a class of graphs. *Discrete Mathematics*, 308:6254–6261, 2008.
- [202] Y. Wu, C.-Q. Zhang, and B.-X. Zhu. Vertex-coloring 3-edge-weighting of some graphs. *Discrete Mathematics*, 340(2):154–159, 2017.
- [203] G. Xavier. *Les Maths en tête, Algèbre*. Les maths en tête, mathématiques pour MP*. Ellipses, Paris, 2009.
- [204] C. Yan, D. Huang, D. Chen, and W. Wang. Adjacent vertex distinguishing edge colorings of planar graphs with girth at least five. *Journal of Combinatorial Optimization*, 28:893–909, 2014.
- [205] M. Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.
- [206] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.
- [207] J. Yue, S. Zhang, and X. Zhang. Note on the perfect EIC-graphs. *Applied Mathematics and Computation*, 289:481 – 485, 2016.
- [208] T. Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4(1):47–74, 1982.
- [209] T. Zaslavsky. Biased graphs. I. Bias, balance, and gains. *Journal of Combinatorial Theory, Series B*, 47(1):32–52, 1989.

- [210] T. Zaslavsky. Biased graphs. II. The three matroids. *Journal of Combinatorial Theory, Series B*, 51(1):46–72, 1991.
- [211] L. Zhang, W. Wang, and K. Lih. An improved upper bound on the adjacent vertex distinguishing chromatic index of a graph. *Discrete Applied Mathematics*, 163(C):348–354, 2014.
- [212] Z. Zhang, L. Liu, and J. Wang. Adjacent strong edge coloring of graphs. *Applied Mathematics Letters*, 15(5):623–626, 2002.
- [213] X. Zhu. Every nice graph is (1,5)-choosable, 2021. <https://arxiv.org/abs/2104.05410>.
- [214] D. Zhuk. A proof of CSP dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, 2017.