Un enfoque algorítmico sobre algunas variantes del problema de coloreo de grafos y el problema de conjunto independiente máximo

Koch, Ivo Valerio

2014-08-21

UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Un enfoque algorítmico sobre algunas variantes del problema de coloreo de grafos y el problema de conjunto independiente máximo

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Ciencias de la Computación

Ivo Valerio Koch

Directora de tesis: Dra. Flavia Bonomo
Director Asistente: Dr. Mario Valencia-Pabon
Consejera de estudios: Dra. Flavia Bonomo

Lugar de trabajo: Instituto de Ciencias, Universidad Nacional de General Sarmiento.

Buenos Aires, Junio de 2014

A Barbi, a mi familia, a mis amigos.

Agradecimientos

A Barbi, porque entre todas las cosas maravillosas que me pasan porque está ella, se encuentra el haber terminado este trabajo.

A mi mamá, a mi papá, a la Oma, al Opa, porque siempre me apoyaron, por el esfuerzo, por el sacrificio, por la inspiración.

A Juan Carlos, a Irusha, a Coni, a Fernando, a Taiel, a Mara, a Virna y a Luis.

A mis compadres Ernst, Alex y a mi ahijado Lucas.

A mis amigos del alma Christian, Eric, Guillermo, Javier, Martin, Rodrigo, Rudy, Tommy y Vito.

A Javier M, por su invalorable ayuda y sus consejos.

A Anibal, por todo lo que me enseñó en la vida profesional.

A Flavia y a Mario, mis directores, por la gran oportunidad de trabajar con ellos, por las magníficas ideas, por los conocimientos enciclopédicos y toda su ayuda.

Un enfoque algorítmico sobre algunas variantes del problema de coloreo de grafos y el problema de conjunto independiente máximo

En esta Tesis estudiamos variantes del problema de coloreo de grafos para varias familias de grafos, y analizamos el problema del conjunto independiente máximo bajo un enfoque de generación de planos de corte.

En el problema del k,i-*coloreo*, asignamos conjuntos de colores de cardinalidad k a los vértices de un grafo G, de manera que los conjuntos que correspondan a vértices adyacentes en G intersequen en no más de i elementos y la cantidad total de colores usados sea mínima. Esta cantidad mínima recibe el nombre de *número* k,i-*cromático* y es denotada por $\chi_k^i(G)$. Este parámetro, que generaliza el número cromático $\chi_1^0(G)$, es tan difícil de trabajar que su valor es desconocido aún para grafos completos. Desarrollamos un algoritmo de orden lineal para el cómputo de χ_k^i para ciclos y generalizamos este resultado a grafos cactus. Adicionalmente, estudiamos la relación entre este problema en grafos completos y un problema abierto clásico de teoría de códigos.

Un *b-coloreo* de un grafo es un coloreo tal que cada clase color admite un vértice adyacente a por lo menos un vértice perteneciente a cada una de las demás clases color. El *número b-cromático* de un grafo G, denotado como $\chi_b(G)$, es el máximo número t tal que G admite un b-coloreo con t colores. Describimos un algoritmo polinomial para computar el número b-cromático de la clase de los grafos P_4-tidy y estudiamos para esta clase dos propiedades conocidas: la *b-continuidad* y la *b-monotonía*.

Estudiamos además la versión sobre aristas del b-coloreo y su *índice b-cromático* asociado. Presentamos cotas para el índice b-cromático del producto directo de grafos y damos resultados generales para varios productos directos de grafos regulares. Introducimos también un modelo sencillo de programación lineal para el b-coloreo de aristas, que utilizamos para calcular resultados exactos para el producto directo de algunas clases de grafos.

Finalmente, proponemos un nuevo método de generación de planos de corte para el problema del conjunto independiente máximo. El algoritmo genera desigualdades de rango y otras desigualdades válidas, y utiliza un procedimiento de lifting basado en la resolución del conjunto independiente máximo con pesos sobre un grafo de menor tamaño.

Palabras clave: k,i-coloreo, grafos cactus, b-coloreo, grafos P_4-tidy, b-coloreo de aristas, producto directo de grafos, conjunto independiente máximo, planos de corte, algoritmos branch and cut.

An algorithmic approach for some variants of the graph coloring problem and the maximum stable set problem

In this Thesis we study variants of the graph coloring problem for several families of graphs, and we address the stable set problem under a new cutting plane generation approach.

In the k, i-*coloring problem*, we assign sets of colors of size k to the vertices of a graph G, so that the sets which belong to adjacent vertices of G intersect in no more than i elements and the total number of colors used is minimum. This minimum number of colors is called k, i-*chromatic number* and is denoted by $\chi_k^i(G)$. This parameter, which generalizes the chromatic number $\chi_1^0(G)$, is so difficult to deal with, that its value is unknown even for complete graphs. We develop a linear time algorithm to compute χ_k^i for cycles and generalize the result to cacti. Further, we study the relation between this problem on complete graphs and a classic open problem in coding theory.

A *b-coloring* of a graph is a coloring such that every color class admits a vertex adjacent to at least one vertex receiving each of the colors not assigned to it. The *b-chromatic number* of a graph G, denoted by $\chi_b(G)$, is the maximum number t such that G admits a b-coloring with t colors. We describe a polynomial time algorithm to compute the b-chromatic number for the class of P_4-tidy graphs and study this class for two known properties: the *b-continuity* and the *b-monotonicity*.

We study also the edge version of the b-coloring problem and its associated *b-chromatic index* for the direct product of graphs and provide general results for many direct products of regular graphs. We introduce a simple linear programming model for the b-edge coloring problem, which we use for computing exact results for the direct product of some special graph classes.

Finally, we propose a general procedure for generating cuts for the maximum stable set problem. The algorithm generates both rank and non-rank valid inequalities, and employs a lifting method based on the resolution of a maximum weighted stable set problem on a smaller graph.

Keywords: k, i-coloring, cacti, b-coloring, P_4-tidy graphs, b-edge-coloring, direct product of graphs, maximum stable set, cutting plane generation, branch and cut algorithms.

Table of Contents

Abstract **i**

1 Introduction **1**

1.1 Historical notes on graph coloring 5
1.2 Definitions and notations 8
1.3 Linear programming and branch and cut algorithms 10
1.4 Resumen del capítulo 12

2 On the k,i-coloring problem of cacti and complete graphs **16**

2.1 New bounds for the k,i-chromatic number 17
2.2 The k,i-coloring problem as graph homomorphism 19
2.3 k,i-coloring of cycles 20
2.3.1 Extension to the $(k:i)$-coloring problem 24
2.3.2 Generalization to cacti 26
2.3.3 Cartesian product of cycles 26
2.4 k,i-coloring of cliques 29
2.5 Resumen del capítulo 33

3 On the b-coloring of P_4-tidy graphs **35**

3.1 Definitions and preliminary results 38
3.2 b-continuity in P_4-tidy graphs 41
3.3 Computation of the b-chromatic number in P_4-tidy graphs 43
3.4 b-monotonicity in P_4-tidy graphs 48
3.5 Resumen del capítulo 51

4 The b-chromatic index of the direct product of graphs **53**

4.1 Bounds for $\chi'_b(G \times H)$ 54
4.2 On the b-chromatic index of direct products of some regular graphs . . 57
4.3 Computing the b-chromatic index by integer linear programming 60
4.4 Direct products of special graph classes 63

4.5 Resumen del capítulo 66

5 A general cut-generating procedure for the stable set polytope 68

5.1 Introduction 68
5.2 The maximum stable set polytope 69
5.3 Clique projection 70
5.4 Clique-Lifting 72
5.5 The cut-separating procedure 74
5.6 Preliminary computational experiments 77
5.7 Resumen del capítulo 80

6 Conclusions 83

6.1 Resumen del capítulo 85

Bibliography 87

Index 94

CHAPTER 1

Introduction

Graph coloring is one of the earliest and most important areas of graph theory. Since its origins in the second half of the nineteenth century, it posed fascinating questions. It gave birth to deep mathematical results, and to many practical applications as well. If graph theory provides a mathematical model for objects involved in a binary relation, graph coloring deals with the fundamental problem of partitioning these objects into classes, according to a set of rules that specify whether any two objects are allowed in the same class. For example, two objects may belong to the same class if they are not related. In the language of graph theory, this means assigning integer numbers ("colors") to the vertices of a graph in such a way that adjacent vertices receive different colors. Such an assignment is called *proper* coloring. We seek also to minimize or maximize some objective function related to the colors. We may want to minimize the total amount of colors used, for instance. This problem is called *classic* graph coloring, and has applications in scheduling, assignment of wavelengths in optical networks, register allocation and timetable design, among others. Let us give an example application.

The number of wireless communications systems deployed around the globe increases continuously. Because of this, the problem of the optimal assignment of a limited radio frequency spectrum becomes of primary importance. The frequency band is normally divided into a number of *channels*. The frequency assignment problem (FAP) models the task of assigning channels to a set of transmitters. A channel can be reused many times for different transmitters if they are far enough apart, so that the co-channel interference is low. If too close transmitters use the same channel simultaneously, the increase of co-channel interference will make the quality of the communication unsatisfactory. Thus it is important to assign adequate channels to the transmitters. Since the available frequency band is limited, we are interested in using as few channels as possible. To model this problem, we define a graph G with one vertex for each transmitter in the network of interest. Two vertices v_{t_1} and v_{t_2} are adjacent if transmitters t_1 and t_2 cannot use the same channel. This graph is called the *conflict* *graph* of the problem. The co-channel FAP is then equivalent to finding a coloring on the graph G with the minimum possible number of colors, each color representing a channel. The coloring represents thus the optimum assignment of channels to the transmitters.

There are real-life problems, however, which resemble a coloring problem but do not match exactly the classic formulation. There could be additional constraints to be satisfied. For this reason, numerous variants of the problem were defined to incorporate these rules to the formulation. Continuing with the above example, suppose the network of interest is now a cellular network. In these kind of networks, the total service region is partitioned into a predefined number of cells, each with a base station located at its center. Mobile users can only communicate with other users via the base station assigned to the cell which they currently occupy. Each base station of a cellular network must be assigned a *set* of channels, say of size k, to be used to communicate with users within its cell region without causing interference to neighboring cells. This transforms the coloring problem we had before into the so called k-*tuple* coloring of the same conflict graph G, where we aim at assigning a set of k colors to each vertex in such a way that adjacent vertices are assigned color sets that do not intersect with each other, seeking to minimize the total amount of colors. See [85] for a complete treatment of this application.

Other graph coloring variants appeared also with a theoretical motivation, such as mathematical generalization. Together, more than 200 different graph coloring problems are studied in the literature, each of them interesting in their own right.

Unfortunately, classic graph coloring as an algorithmic problem is known since the beginning of the 1970s to be NP-complete [69]. As additional constraints are introduced, algorithmic complexity tends to increase even more. One way to work around this difficulty is to restrict our attention to specific graph classes instead of general graphs. The study of some graph coloring problem variants for known graph classes is the main subject of this thesis.

Several generalizations of the coloring problem were introduced in the literature in which each vertex is assigned not only a color but a set of colors, under different restrictions. One of these variations is the k-*tuple coloring* introduced independently by Hilton, Rado and Scott [50], Stahl [92], and Bollobás and Thomason [13]. Brigham and Dutton [18] generalize the concept of k-tuple coloring by introducing the idea of $(k : i)$ *coloring*, in which the sets of colors assigned to adjacent vertices must intersect in exactly i colors. In the k, i-*coloring problem*, this generalization is extended even further: we assign sets of colors of size k to the vertices of a graph G, so that the sets which belong to adjacent vertices of G intersect in no more than i elements and the total number of colors used is minimum. This minimum number of colors is called k, i-*chromatic number* and is denoted by $\chi_k^i(G)$. This variant was introduced by Méndez-Díaz and Zabala in [83]. Very little is known about this problem; even simple graph classes pose a challenge for the k, i-coloring. For example, the computation of χ_k^i is still open for cliques. Our alternative line of work was therefore to examine graph classes that contain only small cliques, and that proved to be fruitful. We present in

Chapter 2 new bounds for the k, i-chromatic number. We develop then a linear time algorithm to compute χ_k^i for cycles and generalize the result to cacti (a graph G is a *cactus* if every edge is part of at most one cycle). We slightly modify the algorithm to solve $(k : i)$-coloring, obtaining a simpler algorithm than the one given by Brigham and Dutton in JCTB [18]. Further, the k, i-chromatic number of some cartesian products is given. Finally, we study the relation between this problem on complete graphs and a classic problem in coding theory. The main part of these results were presented in [15].

The *b-coloring problem* was introduced by Irving and Manlove in 1999 [56]. Given a graph G, we ask for a proper coloring as in the classic coloring, now with the additional requirement that at least one vertex in every color class has in its neighbourhood vertices of all the other colors. We aim here to maximize the number of colors used. This maximum number of colors is called the *b-chromatic number*. The motivation for this coloring comes from the following heuristic for coloring a graph using the minimum number of colors: start from a given coloring and try to decrease the number of colors by eliminating color classes. For this purpose, select (if possible) a color class such that every vertex from that class can be recolored with a different color that is not used by any of its neighbors. Thus, a b-coloring is a coloring where we cannot apply the strategy above to decrease the number of colors. In other words, the b-chromatic number provides an upper bound for the accuracy of this heuristic.

In Chapter 3 we study the b-coloring problem for P_4-tidy graphs. This is a generalization of many classes of graphs with few induced P_4s. We describe in this chapter a polynomial time algorithm to compute the b-chromatic number for this class. For achieving this, we use a decomposition theorem for P_4-tidy graphs due to Giakoumakis et al. [38], based on previous results of Jamison and Olariu [63]. Further, we study for this class two known parameters associated with the b-coloring: the *b-continuity* and the *b-monotonicity*. They are motivated in two aspects in which b-coloring differs from the behavior of classic coloring. It is easy to see that we may always obtain a classic coloring with $k + 1$ colors from a coloring with k colors, as long as $k < V(G)$. This is not necessarily the case for b-colorings. It makes therefore sense to study the graph families in which there exists a t-b-vertex coloring for every integer t between $\chi(G)$ and $\chi_b(G)$. These graph classes are called *b-continuous*. Another atypical property of b-colorings is that the b-chromatic number can increase when taking induced subgraphs. A graph G is *b-monotonous* if $\chi_b(H_1) \geq \chi_b(H_2)$ for every induced subgraph H_1 of G and every induced subgraph H_2 of H_1. We prove that P_4-tidy graphs are b-continuous and b-monotonous.

The results in this chapter extend a previous work by Bonomo et al. [16], where an analogous study was performed on the class of P_4-sparse graphs. Our results were published in [11].

Earlier on we introduced the purpose of colorings as partionining objects into classes. The need for partitioning *pairs* of objects gives rise to *edge colorings*. We illustrate this with a known example from the literature called *timetabling* [19], in this case a Class-Teacher timetabling.

Consider a school having a set of teachers and a set of classes. Every week, a teacher has to teach certain classes. A teacher cannot teach two classes at the same time. Also, a class cannot be taught by two teachers at the same time. The design of the weekly timetable can be modeled as a graph coloring problem as follows: Let G be a bipartite graph where one bipartition class V_t represents the teachers and the other one V_c represent the classes. A vertex $v_t \in V_t$ is connected to a vertex $v_c \in V_c$ if teacher t is in charge of class c. If k is the number of available time slots in the week, the colors correspond to these time slots. A timetable for the teaches and classes is possible if and only if the edges of the bipartite graph can be colored with k colors.

Edge colorings may appear different in nature to vertex colorings, but in fact, they are equivalent to vertex colorings of a special graph class, the so called *line graphs*. The line graph $L(G)$ of G contains one vertex v_e for each edge e of G. Vertices v_e and w_f of $L(G)$ are adjacent if edges e and f share a common vertex in G. In spite of this, the transformation of an instance of the edge coloring problem into its corresponding vertex coloring instance is rarely convenient, because we resign thus the particularities of the edge version in favor of the more general vertex coloring problem. Analogously as with vertex colorings, edge colorings have their own variants that arise from their applications.

We focus in Chapter 4 on the edge coloring version of b-colorings for the direct product of graphs. The *b-chromatic index* $\chi'_b(G)$ of a graph G is the largest integer k such that G has a proper k-edge coloring in which every color class contains at least one edge incident to edges in every other color class. The *direct product* $G \times H$ is the graph with vertex set $V(G) \times V(H)$; two vertices (x, y) and (v, w) are adjacent in $G \times H$ if and only if $xv \in E(G)$ and $yw \in E(H)$.

We give in this chapter bounds for the b-chromatic index of the direct product of graphs and provide general results for many direct products of regular graphs. In addition, we introduce an integer linear programming model for the b-edge coloring problem, which we use for computing exact results for the direct product of some special graph classes. The results in this work were proposed in [74].

In Chapter 5 we move away from colorings and approach another basic graph optimization problem with many applications, the *maximum stable set problem*. Here we do not seek to partition objects into classes, but we aim instead at finding a class of unrelated objects that is as large as possible. The maximum stable set problem is known to be NP-Hard, and was addressed using a number of algorithmic techniques. We work in this chapter with one of these techniques, the integer linear programming approach. Within this technique, a well known family of exact algorithms called *branch and cut* has proven very effective for solving linear programs. One of the building blocks of the branch and cut algorithms is the cutting plane procedure. We propose a general procedure for generating cuts for this problem, inspired by a procedure by Rossi and Smriglio [88] but applying a new lifting procedure by Xavier and Campelo [100]. In contrast to existing cut-generating procedures, our algorithm generates both rank and non-rank valid inequalities, and employs a lifting method based on the solution of a smaller maximum weighted stable set problem. Computational experience on DIMACS benchmark instances shows the competitiveness of the proposed approach. These results were submitted for publication in [29].

Finally, in Chapter 6 we draw conclusions on the results obtained and present possible continuations of this work.

In the rest of this chapter, we present a summary of the historical development of graph coloring. The next section introduces basic definitions and notations that will be used throughout this thesis. We conclude the chapter with an introduction to linear programming and branch and cut algorithms.

1.1 Historical notes on graph coloring

We give here a summary of the origins of the graph coloring problem. We refer the reader to [99] and [26] for a more detailed treatment on this subject. For the most part, both references were the source of the material given here.

On 23 October 1852, English mathematician Augustus de Morgan wrote a letter to his friend Sir William Rowan Hamilton, in which he told him about an intriguing problem that was posed to him by a student of his named Francis Guthrie. His pupil had asked him whether

The regions of every map can be colored with four or fewer colors in such a way that every two regions sharing a common boundary are colored differently.

This problem became known as the Four Color Conjecture, and gave rise to the area of colorability of graphs, that in turn led to the investigation of several other areas of graph theory. It resisted the efforts of mathematicians for over a hundred years, and even its proofs raised enormous controversy, to such extent that the last proof we are aware of dates back to...2007.

Hamilton was not interested in Guthrie's problem, but De Morgan wasn't discouraged and communicated it to many other mathematicians. In April 1860, a book review due to De Morgan appeared in the scientific and literary journal *Atheneum* in which he stated the problem for the first time in print. This contributed to the Conjecture becoming known in the United States. American logician and philosopher Charles Pierce took interest in the problem, and presented an attempted proof to a mathematical society at Harvard University. He also came up with a map drawn on a torus that required six colors.

In 1878, after De Morgan's death, the celebrated mathematician Arthur Cayley asked for a solution in a meeting of the London Mathematical Society, at which he presided. He attacked the problem himself without success and published in the April issue of the *Proceedings of the Royal Geographical Society* in 1879 a short paper in which he tried to explain where the essential difficulty of the problem lied. A former student of Cayley named Alfred Bray Kempe was present at the meeting of 1878. This lawyer and enthusiastic mathematician worked on the problem until 1879, when he announced

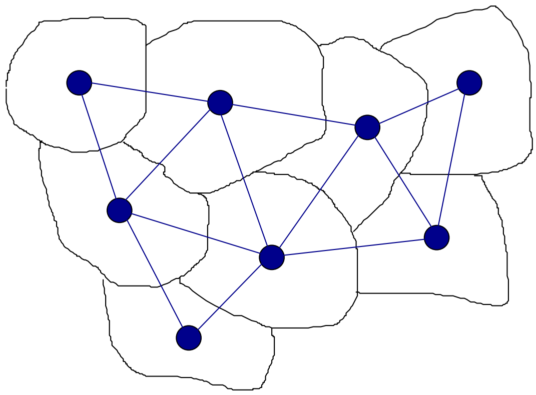
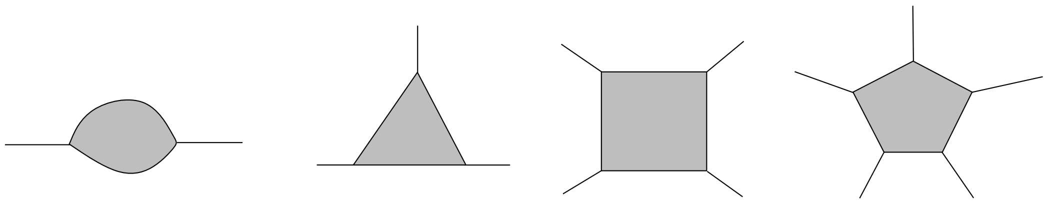


Figure 1.1: A map and its corresponding planar graph. Kempe did not use the word 'graph', but 'linkage'. The word graph was coined by James Sylvester in 1878.

that he had a solution. Cayley encouraged him to submit it to *American Journal of Mathematics*. This way, one of the most famous mistaken proofs in the history of mathematics was published. Kempe's argument certainly contained an error, but some of his ideas were used in subsequent attempts at the problem. Further, he pointed out that the problem could be dualized to coloring the vertices of a graph, as shown in Figure 1.1, introducing thus the modern formulation of the problem. We give here a sketch of his proof. He first proved that every map contained necessarily one of the four configurations illustrated in Figure 1.2. We call such a set *unavoidable*. As a second step, for each configuration he showed that given any map M containing it, and any 4-coloring c of M minus the shaded region of the configuration, c could be extended to a 4-coloring of the whole map. Such a configuration of regions that cannot occur in a minimum counterexample of the Four Color Conjecture is called a *reducible configuration*. It is easy to see that the shaded region in configurations (a) and (b) can be colored with a color not used by its neighbours. For case (c), if the neighbours of the shaded region R use less than 4 colors we are done. If 4 colors are used, Kempe looked at the regions of the map that used two colors. He was able to interchange these two colors so that one of them is 'freed' from the neighbours of R, so we may use it for R. To prove case (d), Kempe used the same argument as in (c), only interchanging *pairs* of colors simultaneously as part of the process. Since all possible cases have been considered, the proof is complete.

For a decade Kempe's theorem was believed to be correct, until in 1890 an English mathematician named Percy Heawood published a paper in which he exposed the flaw in Kempe's work. He presented an example map that couldn't be colored with 4 colors using the method above. Specifically, Kempe's one colour interchanges were always possible, while one cannot always perform two interchanges at the same time. Thus, case (d) was not correctly solved. Nevertheless, Heawood managed to fix enough of Kempe's proof to show that *every map can be colored with five colors.*

The story of the approches to the problem for the next eighty six years is essentially a succession of attempts at constructing unavoidable sets of reducible configurations.



(a) (b) (c) (d)

Figure 1.2: Necessary configurations on any map

Around 1970, German mathematician Heinrich Heesch believed, giving probabilistic arguments, that an unavoidable set of reducible configurations must exist, of cardinality at most 9000. He provided also a technique for constructing unavoidable sets. His ideas were developed by Kenneth Appel and Wolfgang Haken, who spent years programming algorithms that would help in the search for unavoidable configurations, and assist in the reducibility test. They managed thus to produce after 1200 hours of computer time an unavoidable set of 1936 reducible configurations, completing this way the proof of the theorem. Initially, this computer-aided result was resisted by many mathematicians, and raised also interesting questions about the nature of mathematical proof. Since then, the details in the original proof have been simplified, also decreasing the number of configurations, but to the best of our knowledge, all successive improvements are still computer assisted proofs. In 2007, Georges Gonthier published a proof of the theorem using a formal proof system named Coq.

As stated above, already Kempe noted that the map coloring problem is equivalent to coloring the vertices of a planar graph. This modern formulation of the problem was further studied in the 1930s by Hassler Whitney in his Ph.D. thesis, by Rowland Brooks, who in 1941 obtained a good upper bound on the number of colors required, by Gabriel Dirac, who introduced in 1952 the idea of critical graph, and by many others. Whitney developed the idea of the chromatic polynomial of a graph, where the number of possible colorings is a polynomial function of the number of colors available. This polynomial was much studied by George Birkhoff and Bill Tutte, among others.

In 1880, the natural philosopher Peter Guthrie Tait reformulated Kempe's result in terms of the coloring of boundary *edges*, instead of countries, believing that this would simplify the proof. This idea led eventually to the problem of graph edge coloring. Dénes König proved in 1916 that for a bipartite graph G, a number of colors equal to the maximum degree $\Delta(G)$ of G is sufficient. Later Vadim Vizing, in two fundamental papers proved that $\Delta(G)+1$ are always sufficient for any graph G. Nevertheless, the so called *classification problem* of determining whether a graph G needs $\Delta(G)$ or $\Delta(G)+1$ colors is NP-complete [55].

Since the 1970s, the study of coloring problems made great progress, and continues to be one of the most active research areas in graph theory. Further information about this development can be found in [66] and [36].

1.2 Definitions and notations

A *graph* G is an ordered pair $(V(G), E(G))$, that consists of a set $V(G)$ called *vertices* and a set $E(G)$ of pairs of elements of $V(G)$ called *edges* (we will also use the notation $G = (V, E)$). If $V = \emptyset$, G is the *empty graph*. An edge $\{u, v\} \in E(G)$ will be noted uv for simplicity; u and v are called the *endpoints* of the edge. Two vertices connected by an edge are *adjacent*, and two edges that share an endpoint are *incident*. If G allows only one edge between two vertices, it is called *simple*. If multiple edges are allowed, G is called a *multigraph*.

Given a graph G, the *neighborhood* of a vertex v in G is the set of all vertices adjacent to v, denoted by $N_G(v)$ (each of these vertices is a *neighbour* of vertex v). The *closed neighbourhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. The number of neighbours of v is the *degree* of v and is denoted by $d_G(v)$. If all vertices of G have the same degree d, we say that G is a *d-regular graph*. The *minimum degree* of G is the number $\delta(G) = min\{d_G(v), v \in V\}$, and the *maximum degree* of G is $\Delta(G) = max\{d_G(v), v \in V\}$. We may omit the reference to G in the notation when it is clear from context.

The *neighborhood* of an edge e in a graph G is the set $N_G(e)$ of all edges incident to e (the *neighbours* of e), and $N_G[e] = N_G(e) \cup \{e\}$ is its *closed neighborhood*. The *degree* of $e = uv$ is the number of its incident edges $|N_G(e)|$ and is denoted by $d_G(e)$. Note that $d_G(e) = d_G(u) + d_G(v) - 2$. G is called an *r-edge regular graph* if all its edges have the same degree r. Again, the reference to G will be frequently omitted for simplicity.

A *subgraph* of a graph G is another graph H contained in G, such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If H is a subgraph of G and it contains all the edges $(x, y) \in E(G)$, with $x, y \in V(H)$, then H is an *induced subgraph* of G. We say that $V(H)$ *induces* H in G, and write $H = G[V(H)]$.

A graph is *complete* if all its vertices are pairwise adjacent. The complete graph of size n is denoted by K_n. If the subgraph $G[C]$ induced by a set of vertices $C \subseteq V(G)$ is complete, then C is called a *clique* of G.

A subset $S \subseteq V(G)$ is *stable* or *independent* if no two vertices of S are adjacent. The stable set of n vertices is denoted by S_n. The *stability number* of G is the cardinality of a maximum stable set in G and is denoted by $\alpha(G)$.

A graph G is *bipartite* when $V(G)$ can be partitioned into two stable sets.

A *matching* of a graph G is a subset of edges pairwise non-incident.

The *line graph* $L(G)$ of a graph $G = (V, E)$ is the graph having as its vertex set the set E of edges, two vertices in $L(G)$ being adjacent if their corresponding edges in G are incident.

The *complement* graph $\overline{G}$ of G has $V(G)$ as its vertex set, and two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G.

A *walk* is a list $v_0, e_1, v_1, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$ the edge e_i has endpoints v_{i-1} and v_i.
A *path* is a non empty graph $P = (\{v_1, \ldots, v_n\}, \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\})$. The vertices v_1 and v_n are its *endpoints*, and the remaining vertices are *midpoints*. An induced path on n vertices shall be denoted by P_n. An induced subgraph of G isomorphic to P_n is simply said to be a P_n in G. The graph $C = (V(P), E(P) \cup \{(v_n, v_1)\}$ is called a *cycle*. A chordless cycle on n vertices is denoted by C_n. The *length* of a path or cycle is the number of its edges. The *girth* of a graph G is the length of a shortest cycle in G. The *distance* from a vertex u to a vertex v in G, written $d_G(u, v)$ is the length of a shortest path from u to v. If no such path exists, then $d_G(u, v) = \infty$. The *diameter* of a graph G, denoted by $diam(G)$ is $max_{u,v \in V(G)} d(u, v)$.

A graph G is *connected* if there exists a path between any two vertices of G and *disconnected* otherwise. A *connected component* of a graph G is a maximal connected subgraph of G.

If $V' \subseteq V$ and $G = (V, E)$, we write $G - V'$ for $G[V \setminus V']$. If $V' = \{v\}$ is a singleton, we write $G - v$. If $G - v$ has more connected components than G has, then v is a *cut-vertex* of G. A *block* of a graph G is a maximal subgraph of G without cut-vertices. An *end-block* is a block containing exactly one cut-vertex of G.

A connected graph with no cycles is called a *tree*. A graph with no cycles is a *forest*.

A *(proper) vertex coloring* of G (a *vertex coloring*, in short) is an assignment of colors (represented by natural numbers) to the vertices of G, such that any two adjacent vertices are assigned different colors. The smallest number t such that G admits a vertex coloring with t colors is called the *chromatic number* of G and is denoted by $\chi(G)$. A vertex coloring that uses k colors is a k-coloring. A k-coloring is a vertex partition into k independent sets; each of them is called a *color class*.

An *(proper) edge coloring* of a (multi)graph G is an application from the edge set E to a set of colors such that incident edges are assigned different colors. Here, a color class is a matching of G. The minimum number of colors in an edge coloring of G is called the *chromatic index* $\chi'(G)$.

Unless otherwise specified, when we refer simply to a *coloring*, a vertex coloring is to be understood.

Given a k-coloring, a vertex v (resp. edge e) is said to be *dominant* if vertices (resp. edges) receiving all k colors in the coloring can be found in $N[v]$ (resp. $N[e]$). A dominant vertex v (resp. edge e) of color i is called *color i dominating*; we say also that color i is *realized* on v (resp. e).

1.3 Linear programming and branch and cut algorithms

We begin this section by introducing briefly some polyhedral terminology. We follow here mainly the definitions of [90], and refer the reader to this excellent book for a more extensive treatment on the subject.

A subset C of $\mathbb{R}^n$ is *convex* if $\lambda x + (1-\lambda)y$ belongs to C for all $x, y \in C$ and each λ with $0 \leq \lambda \leq 1$. Thus C is convex if it holds for any two points in C that the whole line segment connecting them belongs to C.

The *convex hull* of a set $X \subseteq \mathbb{R}^n$, denoted by $\text{conv}(X)$, is the smallest convex set containing X.

A subset P of $\mathbb{R}^n$ is called a *polyhedron* if there exists an $m \times n$ matrix A and a vector $b \in \mathbb{R}^m$ (for some $m \geq 0$) such that $P = \{x \mid Ax \leq b\}$. A subset P of $\mathbb{R}^n$ is called a *polytope* if it is the convex hull of finitely many vectors in $\mathbb{R}^n$. It can be shown that a subset P of $\mathbb{R}^n$ is a polytope, if and only if it is a bounded polyhedron.

Let $P = \{x \mid Ax \leq b\}$ be a nonempty polyhedron. If c is a nonzero vector for which $\delta = max\{cx \mid x \in P\}$ is finite, then $\{x \mid cx = \delta\}$ is called a *supporting hyperplane* of P. A *face* of P is P itself or the intersection of P with a supporting hyperplane of P. A point x for which $\{x\}$ is a face is called a *vertex* of P. A *facet* of P is an inclusionwise maximal face F of P with $F \neq P$. An inequality determining a facet is called *facet-defining* or *facet-inducing*.

Any linear inequality $c^\top x \leq t$ is called *valid* for P if $c^\top x \leq t$ holds for each $x \in P$.

Let P be a polytope and y a given point. The task of deciding if this point lies in P, and in case it does not, to find a valid inequality for P which is violated by y is called the *separation problem* for polytope P.

Linear programming concerns the problem of maximizing or minimizing a linear function $c^\top x$ over a polyhedron $P = \{x \mid Ax \leq b\}, A \in \mathbb{R}^{m\times n}$ and vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$. A *linear programming problem* (LP) will be denoted by

$$\begin{array}{rll} \text{maximize(minimize)} & c^\top x & \\ \text{subject to} & Ax & \leq b \end{array}$$

or by $max(min)\{c^\top x \mid Ax \leq b\}$. $c^\top x$ is called the *objective function* and the inequalities $Ax \leq b$ are the *constraints*. Frequently an LP is also called *linear programming model*. It can be shown that a minimization problem may be transformed into a maximization problem, so we will consider from this point onwards only the maximization case.

We say that x is an *integer vector* if $x \in \mathbb{Z}^n$. Many combinatorial optimization problems may be formulated as an instance of a linear programming problem, where we seek to optimize an objective function over the integer vectors of some polyhedron P. Such problems are called *integer linear programming problems*, or ILP. While the resolution of LP problems is polynomial [70], solving ILPs is NP-Complete (the satisfiability problem may be transformed into an ILP problem where all variables are binary) [69].

Note that the following inequality holds $max\{c^\top x \mid Ax \leq b, x \text{ integer}\} \leq max\{c^\top x \mid Ax \leq b\}$. This dropping of the integrality constraint is called *linear programming relaxation.* The resulting LP may be used as an upper bound for the ILP.

A standard algorithmic approach for solving ILPs are *branch and cut algorithms*. These are exact algorithms consisting of a combination of a cutting plane method with a branch and bound technique. A brief description of both follows.

A *cutting plane algorithm* receives as input an integer programming formulation of the problem, $max\{c^\top x \mid Ax \leq b, x \text{ integer}\}$. Now, using a linear programming algorithm, it finds an optimal solution x^* to the linear programming relaxation $max\{c^\top x \mid Ax \leq b\}$. If x^* is integral, then it is also an optimal solution to our combinatorial problem. Otherwise, it inspects the valid inequalities to find any inequality that is violated by x^*. The violated inequality is then added to the constraints of the linear programming relaxation. Now a new optimal solution x^{**} is sought and at this point, the procedure is repeated.

In the best scenario, we find this way an integral solution to one of the linear programming relaxations and solve thus the combinatorial problem. But, depending on the objective function and our classes of valid inequalities, the procedure that searches for violations may not find any. In this case, the process terminates without reaching an integral optimal solution. In any case, the optimal value of each linear programming relaxation provides an upper bound on the optimal value of the combinatorial problem that usually improves the previous bound we had.

The *branch and bound method* is a technique for simulating a complete enumeration of all possible solutions without having to consider them one by one. For many NP-hard combinatorial optimization problems, it is the best known framework for obtaining an optimum solution. It consists of the repeated application of a process for splitting the space of solutions into two or more subspaces and applying an upper bounding algorithm to each part. The point of this splitting is that the extra structure in these parts may allow the bounding technique to perform better than on the entire solution space, giving thus an improved upper bound.

Branch and cut algorithms use cutting plane procedures as bounding mechanism in a branch and bound scheme. The branching technique may vary. One common scheme is to choose some variable x_i that takes on a fractional value x_i^* in the optimal solution of the current LP relaxation, and create one new subproblem with the additional constraint $x_i \leq \lfloor x_i^* \rfloor$ and a second subproblem with the additional constraint $x_i \geq \lceil x_i^* \rceil$.

1.4 Resumen del capítulo

Damos en este resumen las definiciones básicas que serán utilizadas a lo largo de todo este trabajo.

Un *grafo* G es un par ordenado $(V(G), E(G))$, que consiste en un conjunto $V(G)$ denominado *vértices* y un conjunto $E(G)$ de pares de elementos de $V(G)$ llamados *aristas* (utilizaremos también la notación $G = (V, E)$). Si $V = \emptyset$, G es el *grafo vacío*. Una arista $\{u, v\} \in E(G)$ será denotada uv por simplicidad; u y v son llamados *extremos* de la arista. Dos vértices conectados por una arista son *adyacentes*, y dos aristas que comparten un extremo son *incidentes*. Si G permite únicamente una arista entre dos vértices, es llamado *simple*. Si se admiten múltiples aristas, G es llamado *multigrafo*.

Dado un grafo G, la *vecindad* de un vértice v en G es el conjunto de vértices adyacentes a v, denotado por $N_G(v)$ (cada uno de estos vértices es un *vecino* de v). La *vecindad cerrada* de v es $N_G[v] = N_G(v) \cup \{v\}$. El número de vecinos de v es el *grado* de v y es denotado $d_G(v)$. Si todos los vértices de G tienen el mismo grado d, decimos que G es un *grafo d-regular*. El *grado mínimo* de G es el número $\delta(G) = min\{d_G(v), v \in V\}$, y el *grado máximo* de G es $\Delta(G) = max\{d_G(v), v \in V\}$. Omitiremos la referencia a G en la notación cuando resulte claro por contexto.

La *vecindad* de una arista e en un grafo G es el conjunto $N_G(e)$ de todas las aristas incidentes a e (los *vecinos* de e), y $N_G[e] = N_G(e) \cup \{e\}$ es su *vecindad cerrada*. El *grado* de $e = uv$ es su número de aristas incidentes $|N_G(e)|$ y es denotado $d_G(e)$. Notar que $d_G(e) = d_G(u) + d_G(v) - 2$. G es llamado *r-arista regular* si todas sus aristas tienen el mismo grado r. Nuevamente, la referencia a G será omitida frecuentemente para simplificar la notación.

Un *subgrafo* de un grafo G es otro grafo H contenido en G, tal que $V(H) \subseteq V(G)$ y $E(H) \subseteq E(G)$. Si H es un subgrafo de G y contiene todas las aristas $(x, y) \in E(G)$, con $x, y \in V(H)$, entonces H es un *subgrafo inducido* de G. Decimos que $V(H)$ *induce* H en G, y lo notamos $H = G[V(H)]$.

Un grafo es *completo* si todos sus vértices son adyacentes de a pares. El grafo completo de tamaño n se denota K_n. Si el subgrafo $G[C]$ inducido por un conjunto de vértices $C \subseteq V(G)$ es completo, entonces C es llamado *clique* de G.

Un subconjunto $S \subseteq V(G)$ es *estable* o *independiente* si ningún par de vértices de S es adyacente. El conjunto estable de n vértices se denota S_n. El *número de estabilidad* de G es la cardinalidad de un conjunto independiente máximo en G y se denota $\alpha(G)$.

Un grafo G es *bipartito* cuando $V(G)$ puede ser particionado en dos conjuntos estables.

Un *matching* de un grafo G es un subconjunto de aristas tal que ningún par de ellas es incidente.

El *grafo de línea* $L(G)$ de un grafo $G = (V, E)$ es el grafo que tiene por vértices el conjunto E de aristas, y tal que dos vértices son adyacentes en $L(G)$ si sus correspondientes aristas en G son incidentes.

El *grafo complemento* $\overline{G}$ de G tiene a $V(G)$ como vértices, y dos vértices son adyacentes en $\overline{G}$ si y sólo si no son adyacentes en G.

Un *camino* es un grafo no vacío $P = (\{v_1, \ldots, v_n\}, \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\})$. Los vértices v_1 and v_n son sus *extremos*, y los vértices restantes son *vértices intermedios*. Un camino inducido de n vértices será denotado P_n. Un subgrafo inducido de G isomorfo a P_n se dice simplemente un P_n en G. El grafo $C = (V(P), E(P) \cup \{(v_n, v_1)\}$ es un *ciclo*. Un ciclo sin cuerdas de n vértices se denota C_n. La *longitud* de un camino o ciclo es su número de aristas. La *cintura* de un grafo G es la longitud de un ciclo de longitud mínima en G. La *distancia* de un vértice u a un vértice v en G, denotada $d_G(u, v)$ es la longitud de un camino de longitud mínima de u a v. Si no existe un camino como se pide, entonces $d_G(u, v) = \infty$. El *diámetro* de un grafo G, denotado $diam(G)$ es $max_{u,v \in V(G)} d(u, v)$.

Un grafo G es *conexo* si existe un camino entre dos vértices cualquiera de G y *no conexo* sino. Una *componente conexa* de un grafo G es un subgrafo conexo maximal de G.

Si $V' \subseteq V$ y $G = (V, E)$, escribimos $G - V'$ para $G[V \setminus V']$. Si $V' = \{v\}$ es un conjunto de un único elemento, escribimos $G - v$. Si $G - v$ tiene más componentes conexas que G, entonces v es un *vértice de corte* de G. Un *bloque* de un grafo G es un subgrafo conexo maximal de G sin vértices de corte. Un *bloque hoja* es un bloque que contiene exactamente un vértice de corte de G.

Un grafo conexo sin ciclos es un *árbol*. Un grafo sin ciclos es un *bosque*.

Un *coloreo (válido) de vértices* de G es una asignación de colores (representados por números naturales) de los vértices de G, tal que dos vértices adyacentes cualquiera tengan asignados colores diferentes. El número t más chico tal que G admite un coloreo de vértices con t colores se llama *número cromático* de G y es denotado como $\chi(G)$. Un coloreo de vértices que utiliza k colores es un k-coloreo. Un k-coloreo es una partición de vértices en k conjuntos independientes; cada uno de ellos es una *clase color*.

Un *coloreo (válido) de aristas* de un (multi)grafo G es una aplicación del conjunto de aristas E a un conjunto de colores de manera tal que aristas incidentes reciban colores diferentes. Aquí, una clase color es un matching de G. El mínimo número de colores en un coloreo de aristas de G es denominado *índice cromático* $\chi'(G)$.

Salvo explícitamente especificado, cuando nos referimos simplemente a un *coloreo*, debe entenderse un coloreo de vértices.

Dado un k-coloreo, un vértice v (resp. arista e) es *dominante* si en $N[v]$ (resp. $N[e]$) se pueden encontrar vértices (resp. aristas) de los k colores del coloreo. Un vértice dominante v (resp. arista dominante e) de color i se denomina *color i dominante*; decimos también que el color i se *realiza* en v (resp. e).

Programación lineal y algoritmos branch and cut

Damos en esta sección una introducción a terminología poliedral. Seguimos aquí mayormente las definiciones de [90], y remitimos a este libro al lector para un tratamiento más extensivo del tema.

Un subconjunto C de $\mathbb{R}^n$ es *convexo* si $\lambda x + (1-\lambda)y$ pertenece a C para todo $x, y \in C$ y cada λ con $0 \leq \lambda \leq 1$. Entonces C es convexo si para dos puntos cualquiera de C el segmento que los conecta pertenece en su totalidad a C.

La *cápsula convexa* de un conjunto $X \subseteq \mathbb{R}^n$, denotada por $\text{conv}(X)$, es el conjunto convexo más pequeño que contenga a X.

Un subconjunto P de $\mathbb{R}^n$ es llamado un *poliedro* si existe una matrix A de $m \times n$ y un vector $b \in \mathbb{R}^m$ (para algún $m \geq 0$) tal que $P = \{x \mid Ax \leq b\}$. Un subconjunto P de $\mathbb{R}^n$ es llamado un *polítopo* si es la cápsula convexa de una cantidad finita de vectores en $\mathbb{R}^n$. Se puede mostrar que un subconjunto P de $\mathbb{R}^n$ es un polítopo si y sólo si es un poliedro acotado.

Sea $P = \{x \mid Ax \leq b\}$ un poliedro no vacío. Si c es un vector distinto de cero para el cual $\delta = max\{cx \mid x \in P\}$ es finito, entonces $\{x \mid cx = \delta\}$ es un *hiperplano de soporte* de P. Una *cara* de P es o bien P mismo o bien la intersección de P con un plano soporte de P. Un punto x para el cual $\{x\}$ es una cara se llama un *vértice* de P. Una *faceta* de P es una cara maximal F de P con $F \neq P$. Se dice de una desigualdad que determina una faceta que *define faceta*.

Una desigualdad lineal $c^\top x \leq t$ es llamada *válida* para P si $c^\top x \leq t$ vale para cada $x \in P$.

Sea P un polítopo e y un punto dado. La tarea de decidir si este punto pertenece a P, y en caso de que no sea así, encontrar una desigualdad para P que es violada por y se llama el *problema de separación* para el polítopo P.

La programación lineal se ocupa del problema de maximizar o minimizar una función lineal $c^\top x$ sobre un poliedro $P = \{x \mid Ax \leq b\}, A \in \mathbb{R}^{m\times n}$ y vectores $b \in \mathbb{R}^m, c \in \mathbb{R}^n$. Un *problema de programación lineal* (PL) será denotado por

$$\begin{aligned} \text{maximizar(minimizar)} \quad & c^\top x \\ \text{sujeto a} \quad & Ax \leq b \end{aligned}$$

o bien como $max(min)\{c^\top x \mid Ax \leq b\}$. $c^\top x$ es la *función objetivo* y las desigualdades $Ax \leq b$ son las *restricciones*. Frecuentemente un PL también es llamado *modelo de programación lineal*. Se puede mostrar que un problema de minimización puede ser transformado en uno de maximización, con lo cual consideraremos a partir de ahora únicamente el caso de maximización.

Decimos que x es un *vector entero* si $x \in \mathbb{Z}^n$. Varios problemas de optimización combinatoria pueden ser formulados como una instancia de un problema de programación lineal, donde buscamos optimizar una función objetivo sobre los vectores enteros de un poliedro P. Estos problemas son llamados *problemas de programación lineal entera*, o PLIs. Mientras la resolución de problemas PL es polinomial [70], resolver PLIs es NP-Completo (el problema de satisfacibilidad puede ser transformado en un PLI donde todas las variables son binarias) [69].

Notar que la siguiente desigualdad es válida $max\{c^\top x \mid Ax \leq b, x \text{ integer}\} \leq max\{c^\top x \mid Ax \leq b\}$. Este abandono de la restricción de integralidad se denomina *relajación lineal*. El PL resultante puede ser utilizado como una cota superior para el PLI.

Un enfoque computacional standard para resolver PLIs son los algoritmos *branch and cut*. Estos son algoritmos exactos que consisten en una combinación de un método de planos de corte con la técnica de branch and bound. A continuación damos una breve descripción de ambos.

Un *algoritmo de generación de planos de corte* recibe como entrada la formulación en programación lineal entera del problema, $max\{c^\top x \mid Ax \leq b, x \text{ integer}\}$. Utilizando un algoritmo para PL, obtiene una solución óptima x^* a la relajación lineal $max\{c^\top x \mid Ax \leq b\}$. Si x^* es entero, entonces también es una solución óptima para nuestro problema combinatorio. Sino se inspeccionan las desigualdades para encontrar alguna que sea violada por x^*. La desigualdad violada es entonces agregada a las restricciones de la relajación lineal. En este punto, se busca una nueva solucion óptima x^{**} y el procedimiento se repite.

En el mejor escenario, encontramos de esta manera una solución entera para una de las relajaciones lineales y resolvemos así el problema combinatorio. Pero, dependiendo de la función objetivo y nuestras clases de desigualdades válidas, el procedimiento que busca violaciones podría no encontrar ninguna. En este caso, el procedimiento termina sin encontrar una solucion entera óptima. De todas formas, el valor óptimo para cada relajación lineal provee una cota superior para el óptimo del problema combinatorio entero, que usualmente mejora la cota previa que teníamos.

El método de *branch and bound* es una técnica para simular una enumeración completa de todas las posibles soluciones sin tener que considerarlas una a una. Para varios problemas de optimización combinatoria que son NP-hard, es el mejor esquema conocido para obtener una solución óptima. Consiste en la aplicación iterativa de un proceso que divide el espacio de soluciones en dos o más subespacios, para aplicar luego un algoritmo que obtenga una cota superior en ambas partes. El objetivo de este trabajo de partición es que la estructura adicional que aporta cada parte puede permitir al algoritmo de cotas un mejor resultado que sobre el espacio total de soluciones.

Los algoritmos branch and cut usan procedimientos de planos de corte como mecanismo para acotar en un esquema de branch and bound. La técnica de branching puede variar. Un esquema común consiste en elegir una variable x_i que toma un valor fraccionario x_i^* en la solución óptima de la relajación lineal actual, y crear un subproblema nuevo con la restricción adicional $x_i \leq \lfloor x_i^* \rfloor$ y un segundo subproblema con la restricción adicional $x_i \geq \lceil x_i^* \rceil$.

CHAPTER 2

On the k, i-coloring problem of cacti and complete graphs

A *k-tuple coloring* of a graph G is an assignment of k colors to each vertex in such a way that adjacent vertices are assigned different colors. The minimum number of colors needed for a k-tuple coloring of G will be noted $\chi_k(G)$. This problem was introduced independently by several authors. Hilton, Rado and Scott [50] used this coloring as an auxiliary concept for studying the so called fractional chromatic number on planar graphs. Stahl [92] studied general properties for k-tuple colorings and found χ_k for bipartite, complete n-partite graphs and cycles. Bollobás and Thomason [13] showed new relations between $\chi_k(G)$ and $\chi(G)$, namely that $min\{\chi_k(G) : \chi(G) = j\} = 2k + j - 2$ and that $min\{\chi_k(G) : G \text{ is uniquely j-colorable}\} = 2k + j - 1$.

Brigham and Dutton [18] generalized the concept of k-tuple coloring by introducing the concept of $(k : i)$-*coloring*, in which the sets of colors assigned to adjacent vertices intersect in exactly i colors. The $(k : i)$-coloring problem consists into finding the minimum number of colors in a $(k : i)$-coloring of a graph G, which we denote by $\chi_k^{(i)}(G)$. Note that $\chi_k^{(0)}(G) = \chi_k(G)$. In this work, the authors gave $\chi_k^{(i)}(G)$ for bipartite graphs and odd cycles. Partial results were also shown for complete graphs.

Another generalization, known as k, i-*coloring*, was introduced by Méndez-Díaz and Zabala in [83], in which the sets of colors assigned to adjacent vertices intersect in at most i colors. Formally, let G be a graph and let k, i, j be non-negative integers, with $0 \leq i \leq k \leq j$. Then a k, i-coloring of G with j colors consists into assigning to each vertex v of G a set $c(v) \subseteq \{1, \ldots, j\}$ of size k such that each pair of adjacent vertices u, v verifies $|c(v) \cap c(u)| \leq i$. These sets are called *color sets*. The minimum positive integer j such that G has a k, i-coloring with j colors is called the k, i-*chromatic number* and is denoted by $\chi_k^i(G)$. Note that for $k = 1$, $i = 0$, we have the classical coloring problem and thus $\chi_1^0(G) = \chi(G)$ for any graph G. For arbitrary k and $i = 0$, we have the k-tuple coloring.

Méndez-Díaz and Zabala solved in [83] the k, i-coloring problem for some values of k and i on complete graphs, studied the notion of perfectness and criticality for the k, i-coloring problem and gave general bounds for the k, i-chromatic number. The authors proposed also an heuristic approach and a linear programming model for the problem, which they further developed and generalized in [84].

Note that $\chi_k^i(G) \leq \chi_k^{(i)}(G)$, since every $(k : i)$-coloring is in particular a k, i-coloring, but they are not necessarily equal, even for complete graphs. We will provide an example in Section 2.4.

We begin this chapter by introducing new bounds on χ_k^i. Next, we present a linear time algorithm to compute the k, i-chromatic number of cycles and generalize the result in order to derive a polynomial algorithm for this problem on cacti. We also show that these results hold for the $(k : i)$-chromatic number of cycles and cacti. Next, we study the k, i-chromatic number for some cartesian products. Finally, we present a relation between the k, i-coloring problem on complete graphs and weighted binary codes. The results in this chapter were submitted for publication in [15].

2.1 New bounds for the k, i-chromatic number

We start the discussion in this section by listing the few existing general bounds for the k, i-chromatic number, all of them due to Méndez-Díaz and Zabala:

Proposition 2.1.1. *[83] The following bounds hold for any graph G:*

i) $\chi_k^i(G) \leq k(max\{\delta(H)/H \text{ induced subgraph of } G\} + 1) - i$.

ii) $\chi_k^i(G) + \chi_k^i(\overline{G}) \leq k(n+1) - 2i, n \geq 3$.

iii) $\chi_k^i(G)\chi_k^i(\overline{G}) \leq \left(\frac{k(n+1)-2i}{2}\right)^2$ *for* $n \geq 3$.

We contribute here with some new bounds, presented in the following paragraphs.

As an immediate corollary of Theorem 2 of [92], we have that if G has an edge, then $\chi_k^{(0)}(G) \leq \chi_{k+1}^{(0)}(G)$. This property does not hold in general for the $(k : i)$- coloring problem. As a counterexample, Brigham and Dutton [18] give $8 = \chi_2^{(1)}(K_7) > \chi_3^{(1)}(K_7) = 7$. We found computationally another counterexample on a smaller graph: it does also hold that $8 = \chi_3^{(2)}(K_6) > \chi_4^{(2)}(K_6) = 7$. Curiously, the strict version of this inequality becomes valid again for k, i-colorings, as we show in the next easy proposition.

Proposition 2.1.2. $\chi_k^i(G) < \chi_{k+1}^i(G)$.

Proof. Let c be any $k+1, i$-coloring of G. We will transform c into a k, i coloring that uses (at least) one less color. For this, remove some color c_1 from every assignment

$c(v), v \in V(G)$. After this operation, some vertices might still have assignments of cardinality $k+1$. For those vertices, simply remove one arbitrary color from their assignments. It is clear that the result is a valid k, i-coloring, since all assignments have size k and since removal of colors leaves intersections either unchanged or smaller in size. □

The reader might wonder from the proof above whether $\chi_k^i(G)$ and $\chi_{k+1}^i(G)$ may indeed differ in more than one. This is effectively the case. An an example, we have $\chi_3^1(K_3) = 6$, but $\chi_4^1(K_3) = 9$ (see Theorem 2.4.1 for a formula that guarantees correctness of these examples).

Brigham and Dutton [18] conjectured that $\chi_k^{(i+1)}(G) \leq \chi_k^{(i)}(G)$ for $0 \leq i \leq k-1$. For $i = 0$ this is true. Their conjecture was disproved two years later by Kelladi and Payan in [71]. Interestingly, the inequality does hold again when shifting to the k, i-coloring problem, as in the previous proposition, and it is strict. We prove this statement below.

Let G be a graph and c a k, i-coloring of G. Let j be a color used in c, and $C_j \subseteq V(G)$ the subset of vertices such that $v \in C_j$ if $j \in c(v)$. Further, for a subset $S \subset V(G)$, define the operation $rep_c(S, j, j')$ that performs a replacement of color j with color j' in every $c(v)$, $v \in S$.

Lemma 2.1.1. *Let c be a $k, i+1$-coloring of G. Let j be a color such that $|c(v) \cap c(w)| \leq i, \forall v \in C_j, w \in V(G), vw \in E(G)$. Let $S \subset C_j$ and j' a color such that $j' \notin \bigcup_{v \in S} c(v)$. Then $rep_c(S, j, j')$ transforms c into another valid $k, i+1$-coloring c' of G. Moreover, c' verifies $|c'(v) \cap c'(w)| \leq i, \forall v \in C'_j, w \in V(G), vw \in E(G)$.*

Proof. We prove first that c' is a valid $k, i+1$-coloring. All color sets of c' have cardinality k, since no *rep* operation was performed on a color set $c(v)$ having $j, j' \in c(v)$. We verify now the intersection size. Suppose we have an edge vw in G such that $|c'(v) \cap c'(w)| > i+1$. Since c was a valid $k, i+1$-coloring of G, either v or w must be one of the vertices affected by the *rep* operation. Let $v \in S$, without loss of generality. We have $|c(v) \cap c(w)| \leq i, \forall w \in V(G)$ by hypothesis, so in c' this intersection couldn't have grown larger than $i+1$.

Let $vw \in E(G)$, with $v \in C'_j, w \in V(G)$. We show now that $|c'(v) \cap c'(w)| \leq i, \forall v \in C'_j, w \in V(G)$. This condition holds for $v \in C'_j, w \in V(G) \setminus S$, because we had by hypothesis $|c(v) \cap c(w)| \leq i, \forall v \in C_j, w \in V(G)$ for every edge vw, and $C'_j \subset C_j$. Let now $v \in C'_j, w \in S$. Then $|c'(v) \cap c'(w)| \leq |c(v) \cap c(w)| \leq i$. To see this, note that $c(v)$ and $c(w)$ shared color j in c, and do not share it in c'. With the replacement of color j with color j', $c'(v)$ and $c'(w)$ could possible share now j', so the cardinality of $c'(v) \cap c'(w)$ is at most the size of $c(v) \cap c(w)$. □

Proposition 2.1.3. *Let G be a graph with at least one edge, and let $k > i$. Then $\chi_k^{i+1}(G) < \chi_k^i(G)$.*

Proof. Let c be a k, i-coloring of G. We will transform c into a $k, i+1$ coloring that uses one less color. Our strategy will be to remove an arbitrary color j of c by repeatedly applying Lemma 2.1.1. For this purpose, find first a subset $S \subset C_j$ and j' a color such that $j' \notin \bigcup_{v \in S} c(v)$. If such S and j' cannot be found, it would mean that every color j' is present in $\bigcup_{v \in S} c(v)$, for every possible subset $S \subset C_j$. This would imply that only k colors are present in the coloring. If this is the case, c could only be a valid coloring if G has no edges, or if $k = i$. Both situations contradict our hypothesis. Let thus be S and j' as asked. Perform now $rep_c(S, j, j')$. It is easy to see that we are in the hypothesis of Lemma 2.1.1, so the result is a valid $k, i+1$-coloring, such that $C'_j \subsetneq C_j$ and such that the hypothesis of the Lemma still holds. Repeat these two steps while possible. The argument for the existence of S and j', along with C'_j being strictly contained in C_j guarantees that the procedure stops when color j is effectively eliminated from all color sets, and the proposition is proven. □

The following proposition is straightforward for $(k : i)$ colorings, and extends also easily to k, i-colorings.

Proposition 2.1.4 (From Lemma 1 in [18])**.** *Let G be a graph. Then $\chi^i_k(G) \leq \chi^{i-r}_{k-s}(G) + \chi^r_s(G)$.*

A Corollary for this proposition can also be found in the same article. The proof given there works also for k, i-colorings.

Corollary 2.1.1.1 (From Corollary 1 in [18])**.** *Let G be a graph with n vertices. Then $\chi^i_k(G) + \chi^i_k(\overline{G}) \leq k(n+1) - i(n-1)$.*

Note that this improves the similar bound given in Proposition 2.1.1, item *ii*.

2.2 The k, i-coloring problem as graph homomorphism

We'll briefly examine in this section the k, i-coloring problem under another point of view. A graph G is t-colorable if and only if there is a graph homomorphism from G to the complete graph on t vertices K_t, where an *homomorphism* from a graph G to a graph H is an edge preserving map between G and H. Denley [31] introduced the *generalized Kneser graphs* $K(j, k, i)$ as follows. Let i, j, k be integers such that $0 \leq i \leq k \leq j$. Define the graph $K(j, k, i)$ as the graph having as set of vertices the family of k-subsets of $\{1, \ldots, j\}$, and where two k-subsets A and B are adjacent if and only if $|A \cap B| \leq i$. When $i = 0$, the graphs $K(j, k, 0)$ are the well known Kneser graphs [39]. It is not difficult to see that a graph G admits a k, i-coloring with j colors if and only if there is a graph homomorphism from G to $K(j, k, i)$.

As a natural generalization of a theorem due to Harary, Hedetniemi and Prins [47] on the chromatic number, we provide the following

Theorem 2.2.1. *If $\gamma : G \to H$ is a homomorphism, then $\chi^i_k(G) \leq \chi^i_k(H)$.*

Proof. Let $j = \chi^i_k(H)$. Then there exists a homomorphism α from H to $K(j, k, i)$. Since the composition (denoted by $\circ$) of two homomorphisms is again a homomorphism, we have an homomorphism $\gamma \circ \alpha$ from G to $K(j, k, i)$. Hence it follows that $\chi_k^i(G) \leq j = \chi_k^i(H)$. □

Corollary 2.2.1.1. *If graphs G and H have homomorphisms $\alpha : G \to H$ and $\beta : H \to G$, then $\chi_k^i(G) = \chi_k^i(H)$*

Stahl provided similar results for k-tuple colorings in [92].

2.3 k, i-coloring of cycles

Multicycles are cycles in which we can have parallel edges between two consecutive vertices. A multigraph is *k-uniform* if the number of parallel edges between any two adjacent vertices is exactly k.

Let G be a (multi)cycle on n vertices, $m \geq n$ edges and maximum degree equal to Δ. It is well known that $\chi'(G) = \Delta$ if n is even. In fact, it follows from König's Theorem on edge-coloring of bipartite (multi)graphs. When n is odd, we have the following result due to Berge.

Theorem 2.3.1. *[10] Let $G = (V, E)$ be a multicycle on n vertices with m edges and maximum degree Δ. Let $\tau = \lfloor \frac{n}{2} \rfloor$ denote the maximum cardinality of a matching in G. Then*

$$\chi'(G) = \begin{cases} \Delta & \text{if } n \text{ is even,} \\ max\{\Delta, \lceil \frac{m}{\tau} \rceil\} & \text{if } n \text{ is odd} \end{cases}$$

Let G be a k-uniform multicycle on n vertices. It is not difficult to see that the line graph $L(G)$ of G can be seen as the cycle C_n where each vertex is replaced by a clique of size k and all edges between two disjoint copies of K_k associated with two adjacent vertices in C_n are added. Therefore, we can rephrase Theorem 2.3.1 for k-uniform multicycles in terms of a vertex coloring problem of $L(G)$ as follows.

Corollary 2.3.1.1. *Let $L(G)$ be the line graph of a k-uniform multicycle G on n vertices. Let $\alpha = \lfloor \frac{n}{2} \rfloor$ denote the maximum cardinality of an independent set in $L(G)$. Then*

$$\chi(L(G)) = \begin{cases} 2k & \text{if } n \text{ is even,} \\ max\{2k, \lceil \frac{nk}{\alpha} \rceil\} & \text{if } n \text{ is odd} \end{cases}$$

Corollary 2.3.1.1 has been obtained independently by Stahl [92].

By using a theorem of Stahl ([92], p.193), Brigham and Dutton obtain the following result on cycles:

Theorem 2.3.2. *[18] Let C_n be a cycle, with $n = 2t + 1$. Then,*

$$\chi_k^{(i)}(C_n) = \begin{cases} 2k - i & \text{if } k \leq i(t+1), \\ 2k - i + 1 + \lfloor \frac{k - i(t+1) - 1}{t} \rfloor = \lceil \frac{n(k-i)}{t} \rceil & \text{if } k > i(t+1). \end{cases}$$

However, it is not so evident how to construct efficiently a $(k : i)$-coloring of an odd cycle C_n with $\chi_k^{(i)}(C_n)$ colors in polynomial time.

It was already noticed in [83] that a bipartite graph has k, i-chromatic number at most $2k - i$, and that this is also the trivial lower bound for the k, i-chromatic number of any graph with at least one edge. Since even cycles are bipartite, this case is solved, and we will turn our attention to the odd case. In this section, we obtain a similar result as the one found by Brigham and Dutton [18] on odd cycles (Theorem 2.3.2). We prove that the k, i-chromatic number and the $(k : i)$-chromatic number are equal on odd cycles. Furthermore, we derive a simple linear time algorithm to k, i-color an odd cycle with the minimum number of colors, and we adapt it also for $(k : i)$-coloring.

We will compute first a lower bound for the k, i-chromatic number of C_n as follows.

Lemma 2.3.3. *Let C_n be a cycle on $n = 2t + 1$ vertices. Then, for any non-negative integers i, k with $0 \leq i \leq k$, we have that : $\chi_k^i(C_n) \geq \max\{2k - i, \lceil \frac{n(k-i)}{t} \rceil\}$.*

Proof. Notice that $2k - i$ is a trivial lower bound for any graph with at least one edge. So, we only need to prove that $\chi_k^i(C_n) \geq \lceil \frac{n(k-i)}{t} \rceil$, where $n = 2t + 1$. Assume that the vertices of C_n are labeled consecutively by $v_0, \ldots, v_{n-1}$. Arithmetic operations will be taken modulo n. Let c be an optimum k, i-coloring of the vertices of C_n, that is, for each vertex v_i we have that $|c(v_i)| = k$; for each pair of adjacent vertices v_i, v_{i+1} we have that $|c(v_i) \cap c(v_{i+1})| \leq i$; and the maximum color used by c is equal to χ_k^i. Now, for each vertex v_i in C_n, let $c'(v_i) = c(v_i) \setminus (c(v_i) \cap c(v_{i+1}))$. Notice that the size of each set $c'(v_i)$ is at least $k - i$, and that $c'(v_i) \cap c'(v_{i+1}) = \emptyset$ for every $i = 1, \ldots, n$. Therefore, it is not difficult to deduce that the sets c' can be used in order to color the vertices of the line graph of a multicycle on n vertices having at least $k - i$ parallel edges between each pair of adjacent vertices. By Corollary 2.3.1.1, the result follows. ☐

Now, in order to compute an upper bound for the k, i-chromatic number of cycles, we will construct a k, i-coloring for these graphs. First, we need the following lemma.

Lemma 2.3.4. *Let n, n' be two odd integers, with $n' > n \geq 3$. Then any k, i-coloring of C_n can be extended to a k, i-coloring of $C_{n'}$ without using additional colors.*

Proof. Let $v_1, \ldots, v_n$ be the vertices of C_n and let c be a k, i-coloring of C_n. Let $v'_1, \ldots, v'_{n'}$ be the vertices of $C_{n'}$ and define c' as $c'(v'_i) = c(v_i)$ for $i = 1, \ldots, n$; $c'(v_{n+j}) = c(v_{n-1})$ if j is odd, $c'(v_{n+j}) = c(v_n)$ if j is even, for $j = 1, \ldots, n' - n$. It is easy to check that c' is a k, i-coloring of $C_{n'}$. ☐

Based on this, we propose the following simple algorithm.

Lemma 2.3.5. *Let $n = 2t + 1$ with $t \geq 1$. Then, $\chi_k^i(C_n) \leq \max\{2k - i, \lceil \frac{n(k-i)}{t} \rceil\}$. Moreover, a k, i-coloring of C_n with $\max\{2k - i, \lceil \frac{n(k-i)}{t} \rceil\}$ colors can be obtained by Algorithm 1.*

Algorithm 1

Input: A cycle C_n, $n = 2t+1$, with vertices $v_1, v_2, \ldots, v_n$, integers k and i with $0 \leq i < k$.

Output: An assignment c of k colors from $[1, \max\{2k-i, \lceil \frac{n(k-i)}{t} \rceil\}]$ to each vertex of C_n.

1: Let $N = \max\{2k-i, \lceil \frac{n(k-i)}{t} \rceil\}$; $\ell = 1$. Let t' be the minimum positive integer value such that $\lceil \frac{(2t'+1)(k-i)}{t'} \rceil = \lceil \frac{n(k-i)}{t} \rceil$, i.e., either $t' = 1$ or $t' > 1$ and $\lceil \frac{(2t'-1)(k-i)}{t'-1} \rceil > \lceil \frac{n(k-i)}{t} \rceil$. (This value can be obtained by binary search.)

2: For $j = 1$ to $2t'+1$ do:
  If $\ell + k - 1 \leq N$ then
   $c(v_j) = [\ell, \ell + k - 1]$
  else
   $c(v_j) = [\ell, N] \cup [1, \ell + k - 1 - N]$
  end if
  If $\ell + k - i \leq N$ then
   $\ell = \ell + k - i$
  else
   $\ell = \ell + k - i - N$
  end if
 end for

3: For $j = t'+1$ to t do:
  $c(v_{2j}) = c(v_{2t'})$
  $c(v_{2j+1}) = c(v_{2t'+1})$
 end for

Proof. Let us see that the assignment c obtained by Algorithm 1 on C_{2t+1} defines a k, i-coloring.

Note that the algorithm assigns circular intervals of size k (i.e., either intervals of k consecutive numbers or intervals formed by the last d and the first $k-d$ numbers) in such a way that $c(v_1) = \{1, 2, \ldots, k\}$ and for $2 \leq j \leq 2t'+1$, $c(v_j)$ is the circular interval whose first i colors are the last i colors of $c(v_{j-1})$. As we have at least $2k-i$ colors, the intersection of $c(v_j)$ and $c(v_{j-1})$ are exactly those i colors. The property $|c(v_j) \cap c(v_{j-1})| = i$ holds also for $2t'+2 \leq j \leq 2t+1$, when $t' < t$, since they use alternately $c(v_{2t'})$ and $c(v_{2t'+1})$. Therefore, in order to ensure that c is a valid k, i-coloring of C_{2t+1}, we just need to check that $|c(v_{2t'+1}) \cap c(v_1)| \leq i$.

By construction, the first number in the circular interval $c(v_{2t'+1})$ is the number d in $[1, N]$ that is congruent to $2t'(k-i)+1$ modulo N. We should prove

$$k - i + 1 \leq d \leq N - (k-i) + 1.$$

If $t' = 1$, then $2t'(k-i)+1 = 2(k-i)+1$ and it holds $k-i+1 \leq 2(k-i)+1$. Also, $2(k-i)+1 \leq N-(k-i)+1$ if and only if $3(k-i) \leq N$, but $N = \max\{2k-i, \lceil \frac{n(k-i)}{t} \rceil\}$ and $\lceil \frac{n(k-i)}{t} \rceil = \lceil \frac{(2t'+1)(k-i)}{t'} \rceil = 3(k-i)$, so $d = 2(k-i)+1$ and this finishes the case

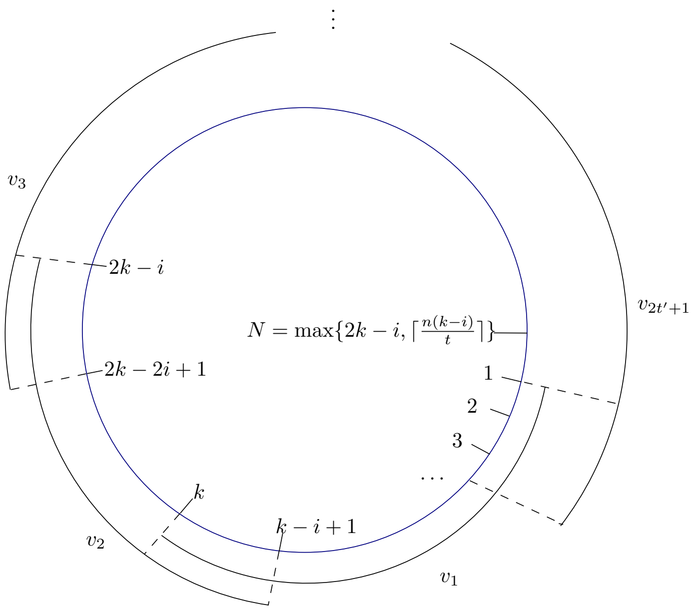


Figure 2.1: Illustration of the k, i-coloring algorithm. We assume colors $\{1, 2, \dots, N\}$ arranged consecutive around the circle. Recall that from vertices $v_{2t'+2}$ onwards we repeat alternatively the assignments of $v_{2t'}$ and $v_{2t'+1}$. The dashed lines represent an intersection of i colors in all arcs except possibly $v_{2t'+1}$ with v_1.

$t' = 1$. Assume from now on that $t' > 1$ and $\lceil\frac{n(k-i)}{t}\rceil = \lceil\frac{(2t'+1)(k-i)}{t'}\rceil$ but $\lceil\frac{n(k-i)}{t}\rceil < \lceil\frac{(2t'-1)(k-i)}{t'-1}\rceil$, so $\lceil\frac{n(k-i)}{t}\rceil < \frac{(2t'-1)(k-i)}{t'-1}$. We will split now the proof into two cases, depending on the value of N.

Case 1: $N = 2k-i$. Note that $\lceil\frac{n(k-i)}{t}\rceil = \lceil\frac{(2t+1)(k-i)}{t}\rceil = \lceil\frac{2tk-2ti+k-i}{t}\rceil = 2k-i+\lceil\frac{(k-(t+1)i)}{t}\rceil$. So, $\lceil\frac{n(k-i)}{t}\rceil \le 2k-i \Leftrightarrow \frac{(k-(t+1)i)}{t} \le 0 \Leftrightarrow k \le (t+1)i$. In particular, this will not be the case if $i = 0$. Thus, $\max\{2k-i, \lceil\frac{n(k-i)}{t}\rceil\} = 2k-i$ if and only if $i > 0$ and $\frac{k}{i} \le t+1 \Leftrightarrow \lceil\frac{k}{i}\rceil - 1 \le t$. By our assumption about t' and as we have discarded the case $t' = 1$, it should be $t' = \lceil\frac{k}{i}\rceil - 1$.

But $2t'(k-i)+1 = t'(2k-i)-t'i+1 \equiv 2k-i-t'i+1 \pmod{2k-i}$. Since $t'i < k \le (t'+1)i$, it holds $k-i+1 < k-i+k-t'i+1 = k+k-(t'+1)i+1 \le k+1$, so $d = 2k-i-t'i+1$ and this closes Case 1.

Case 2: $N = \lceil\frac{n(k-i)}{t}\rceil = \lceil\frac{(2t'+1)(k-i)}{t'}\rceil$. By the analysis in Case 1, that means $\frac{(k-(t'+1)i)}{t'} > 0$. Let $b = \lceil\frac{(k-(t'+1)i)}{t'}\rceil$, thus $N = 2k-i+b$. By our assumption about t' and as we have discarded the case $t' = 1$, it should be $\frac{(k-t'i)}{t'-1} > b$.

In this case, $2t'(k-i)+1 = t'N - t'i - t'b + 1 \equiv N - t'i - t'b + 1$ modulo N. On one hand,

$$\begin{aligned} N - t'i - t'b + 1 &\leq N - (k-i) + 1 \quad \Leftrightarrow \\ -t'(i+b) &\leq -(k-i) \quad \Leftrightarrow \\ \frac{k-(t'+1)i}{t'} &\leq b \end{aligned}$$

and this is satisfied because $b = \lceil \frac{(k-(t'+1)i)}{t'} \rceil$. On the other hand,

$$\begin{aligned} k - i + 1 &\leq N - t'i - t'b + 1 = (2k - i + b) - t'(i+b) + 1 \quad \Leftrightarrow \\ (t'-1)b &\leq k - t'i \quad \Leftrightarrow \\ b &\leq \frac{(k-t'i)}{t'-1} \end{aligned}$$

and we have observed that this inequality already holds. So $d = N - t'i - t'b + 1$ and this ends the proof of this lemma. □

By the proofs of Lemmas 2.3.3 and 2.3.5, we have the following result.

Theorem 2.3.6. *Let C_n be a cycle on $n = 2t+1$ vertices. Then, $\chi_k^i(C_n) = max\{2k - i, \lceil \frac{n(k-i)}{t} \rceil\}$ and a k, i-coloring of C_n with $\chi_k^i(C_n)$ colors can be obtained in $O(n)$ time.*

For example, the $4,1$-coloring of C_3 obtained by Algorithm 1 is $\{1,2,3,4\}$, $\{4,5,6,7\}$, $\{7,8,9,1\}$, the $4,1$-coloring of C_5 obtained by Algorithm 1 is $\{1,2,3,4\}$, $\{4,5,6,7\}$, $\{7,8,1,2\}$, $\{2,3,4,5\}$, $\{5,6,7,8\}$, the $4,1$-coloring of C_7 obtained by Algorithm 1 is $\{1,2,3,4\}$, $\{4,5,6,7\}$, $\{7,1,2,3\}$, $\{3,4,5,6\}$, $\{6,7,1,2\}$, $\{2,3,4,5\}$, $\{5,6,7,1\}$, and the $4,1$-coloring of C_{11} obtained by Algorithm 1 is an extension of the coloring of C_7, namely, $\{1,2,3,4\}$, $\{4,5,6,7\}$, $\{7,1,2,3\}$, $\{3,4,5,6\}$, $\{6,7,1,2\}$, $\{2,3,4,5\}$, $\{5,6,7,1\}$, $\{2,3,4,5\}$, $\{5,6,7,1\}$, $\{2,3,4,5\}$, $\{5,6,7,1\}$.

2.3.1 Extension to the $(k:i)$-coloring problem

Note that an optimal k, i-coloring of C_{2t} is always a $(k:i)$-coloring, since it uses $2k-i$ colors, but for odd cycles this is not always the case. Indeed, the $4,1$-coloring of C_5 obtained by Algorithm 1 is not a $(4:1)$-coloring, since $c(v_5) \cap c(v_1) = \emptyset$.

Note also that an analogous to Lemma 2.3.4 can be proved for the $(k:i)$-coloring problem. We will show now that, if a k, i-coloring c of C_{2t+1} is obtained by Algorithm 1, one can modify the set $c(v_{2t+1})$ by a simple procedure, in order to obtain a $(k:i)$-coloring of C_{2t+1} with the same number of colors.

First notice that $|c(v_i) \cap c(v_{i+1})| = i$ for $i = 1, \ldots, 2t$, and $|c(v_{2t+1}) \cap c(v_1)| \leq i$. Assume $|c(v_{2t+1}) \cap c(v_1)| < i$, otherwise we are done. We have to show how to increase $|c(v_{2t+1}) \cap c(v_1)|$ without decreasing $|c(v_{2t+1}) \cap c(v_{2t})|$.

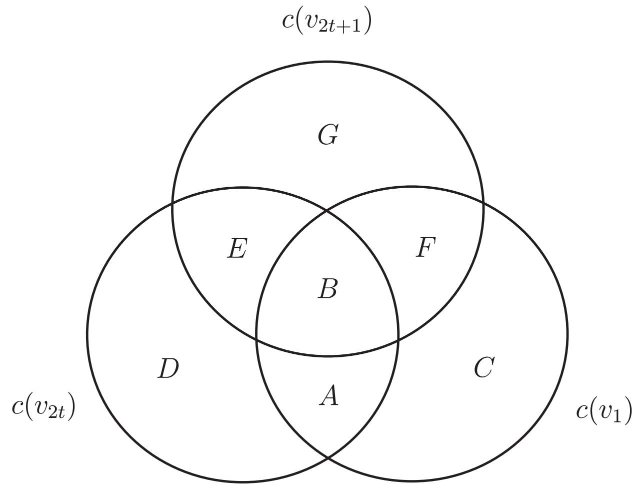


Figure 2.2: Diagram for the definition of color sets.

Let us define the following sets: $A = c(v_1) \cap c(v_{2t}) \setminus c(v_{2t+1})$, $B = c(v_1) \cap c(v_{2t}) \cap c(v_{2t+1})$, $C = c(v_1) \setminus (c(v_{2t}) \cup c(v_{2t+1}))$, $D = c(v_{2t}) \setminus (c(v_1) \cup c(v_{2t+1}))$, $E = c(v_{2t}) \cap c(v_{2t+1}) \setminus c(v_1)$, $F = c(v_1) \cap c(v_{2t+1}) \setminus c(v_{2t})$, $G = c(v_{2t+1}) \setminus (c(v_1) \cup c(v_{2t}))$ (see Figure 2.2), and let $x = |X|$ for $X = A, \ldots, G$.

If $g > 0$ and $c > 0$, we can replace in $c(v_{2t+1})$ a color from G by a color from C, and if $e > 0$ and $a > 0$, we can replace in $c(v_{2t+1})$ a color from E by a color from A. In both cases, we are increasing $|c(v_{2t+1}) \cap c(v_1)|$ without decreasing $|c(v_{2t+1}) \cap c(v_{2t})|$.

If $c = 0$, the total number of colors used by v_1, v_{2t}, and v_{2t+1} is $2k - i$, so $|c(v_{2t+1}) \cap c(v_1)| \geq i$, a contradiction to our assumption. So, $c > 0$. If $g = 0$ then $e > 0$, otherwise $|c(v_{2t+1})| = b + f < i \leq k$, a contradiction. Therefore, we only have to show that if $g = 0$ then $a > 0$. Suppose $g = a = 0$. Then $c > k - i$, $d = k - i$, and $b + e + f = k$. So, the total number of colors used by v_1, v_{2t}, and v_{2t+1} is strictly greater than $3k - 2i$. We will show that, instead, the number of colors used by Algorithm 1 is at most $3k - 2i$. It is clear that $2k - i \leq 3k - 2i$ since $i \leq k$, so we will assume that the number of colors used is $2k - i + \lceil \frac{(k-(t+1)i)}{t} \rceil$.

$$2k - i + \lceil \frac{(k - (t+1)i)}{t} \rceil \leq 3k - 2i \;\Leftrightarrow\; \lceil \frac{(k - (t+1)i)}{t} \rceil \leq k - i \;\Leftrightarrow$$
$$\frac{(k - (t+1)i)}{t} \leq k - i \;\Leftrightarrow\; 0 \leq (t-1)k + i$$

And this completes the argument.

In the previous example, the $4, 1$-coloring of C_5 obtained by Algorithm 1 would be modified as to obtain, for instance, the following $(4:1)$-coloring: $\{1, 2, 3, 4\}$, $\{4, 5, 6, 7\}$, $\{7, 8, 1, 2\}$, $\{2, 3, 4, 5\}$, $\{5, 6, 7, 1\}$.

It may be interesting to characterize in general the graphs G such that $\chi_k^i(G) = \chi_k^{(i)}(G)$, or those graphs G such that $\chi_k^i(H) = \chi_k^{(i)}(H)$ for each induced subgraph H of G.

2.3.2 Generalization to cacti

These results can be easily generalized for cacti. A graph G is a *cactus* if it does not contain two cycles that share an edge. It is a known fact that every block of a cactus is either an edge or a chordless cycle. We will base our proof on the following easy lemma, that holds for many coloring problems.

Lemma 2.3.7. *Let G be a graph. The k, i-chromatic number of G is the maximum of the k, i-chromatic numbers of its blocks.*

Proof. Clearly, it is enough to prove it for connected graphs. We proceed by induction on the number of blocks m of G. If G has only one block, the result trivially holds. For the inductive case, suppose the lemma holds for all graphs with fewer than m blocks. Let B be an end-block of G and let v be the cut-vertex of G that belongs to B. Let G' be the subgraph of G induced by $(V(G) \setminus B) \cup \{v\}$. By inductive hypothesis, the k, i-chromatic number of G' is the maximum of the k, i-chromatic numbers of its blocks.

Let f' be a k, i-coloring of G' with the minimum number of colors, and f'' be an optimal k, i-coloring of the subgraph of G induced by B. By renaming the colors in f'' in such a way that $f''(v) = f'(v)$, we can combine f' and f'' in order to obtain a k, i-coloring of G without adding any new colors. This proves the lemma. □

By Theorem 2.3.6 and Lemma 2.3.7, we obtain directly the following result.

Corollary 2.3.7.1. *Let G be a cactus. Then, a k, i-coloring of G with $\chi_k^i(G)$ colors can be computed in linear time.*

Note that Lemma 2.3.7 and Corollary 2.3.7.1 can be proved analogously for the $(k : i)$-coloring problem.

2.3.3 Cartesian product of cycles

The *cartesian product* of two graphs G and H, denoted by $G \square H$, is the graph with vertex set $V(G) \times V(H)$ specified by making (u, v) adjacent to (u', v') if and only if $u = u'$ and $vv' \in E(H)$, or $v = v'$ and $uu' \in E(G)$. Given a vertex $v \in V(H)$, the subgraph G_v of $G \square H$ induced by $\{(u, v) : u \in V(G)\}$ is called a *G-fiber*; H-fibers are defined similarly. See Figure 2.3 for an illustration on the cartesian product of two cycles.

A well known theorem due to Vizing settles the chromatic number for the cartesian product of two graphs:

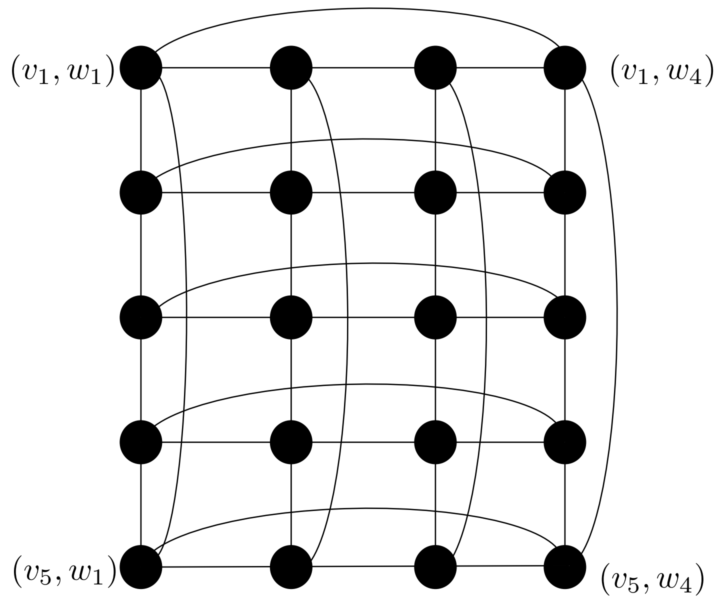


Figure 2.3: Cartesian product $C_5 \square C_4$. The first column of vertices, the subgraph $C_{5_{w_1}}$ is a C_5-fiber and the first row, the subgraph $C_{4_{v_1}}$ is a C_4-fiber.

Theorem 2.3.8. *[97]* $\chi(G \square H) = max\{\chi(G), \chi(H)\}$

We tried to produce a similar result for the k, i-coloring problem, but were not able to prove or disprove the general case. Instead, we offer below versions of the theorem for some particular products.

Proposition 2.3.1. *If G and H are bipartite graphs, then* $\chi_k^i(G \square H) = max\{\chi_k^i(G), \chi_k^i(H)\}$.

Proof. Immediate from the fact that the cartesian product of two bipartite graphs is bipartite. Clearly, if one of H and G has at least one edge, then $\chi_k^i(G \square H) = 2k - i$. □

Proposition 2.3.2. *If* $\chi(G) = j$, *then* $\chi_k^i(G \square K_j) = max\{\chi_k^i(G), \chi_k^i(K_j)\}$.

Proof. Let c be a classic coloring of G using j colors, and c' a k, i-coloring of K_j with $\chi_k^i(K_j)$ colors. For notational simplicity, we will consider colors starting from 0, as well as labels on vertices. Denote by J the j color sets of c' (there must be indeed j different color sets unless $k = i$, but in this case the proposition holds trivially). Let $f : \{0, 1, \ldots, j-1\} \to J$ be a bijective function from each color c_i in c to a color set in c'. Observe that $f(c(v)), v \in V(G)$ configures a valid k, i-coloring of G using $\chi_k^i(K_j)$ colors, so $\chi_k^i(G) \leq \chi_k^i(K_j)$. Let also $F_j = \{f_0, f_2, \ldots, f_{j-1}\}$ be a family of bijections, $f_t : \{0, 1, \ldots, j-1\} \to \{0, 1, \ldots, j-1\}, 0 \leq t < j$, defined by:

$$f_t(r) = (r + t) \pmod{j}$$

Observe that no two bijections in F_j map an r to the same value. To see this, suppose that $0 \leq t < t' < j$ and that $f_t(r) = f_{t'}(r)$. Hence, $(r + t) \equiv (r + t') \pmod{j}$. But this

implies that $t \equiv t' \pmod{j}$, and this cannot hold since $0 \leq t < t' < j$.
Perform now for each vertex w of each fiber G_t of $G\Box K_j$ the assignment $c''(w) = f(f_t(c(w)))$. We will show that c'' is a valid k, i-coloring of $G\Box K_j$ by examining the possible edges. Let $(u, v)(u', v) \in E(G\Box K_j)$. Notice that $uu' \in E(G)$, and vertices (u, v) and (u', v) belong to the same G-fiber G_v. Since $uu' \in E(G)$, $c(u) \neq c(u')$. f_v and f are both injective, so we have $f_v(c(u)) \neq f_v(c(u'))$ and $f(f_v(c(u))) \neq f(f_v(c(u')))$. Thus (u, v) and (u', v) are assigned different color sets of J, and hence their intersection size is i or less.
Let now $(u, v)(u, v') \in E(G\Box K_j)$. We have $vv' \in K_j$ and (u, v) and (u, v') belong to different G fibers, namely $G_{v'}$ and G_v respectively. Since f_v and $f_{v'}$ map all colors to different values, we have $f_{v'}(c(v)) \neq f_v(c(v))$. Again by injectivity of f, we have $f(f_{v'}(c(v))) \neq f(f_v(c(v)))$. This way (u, v) and (u, v') are assigned different color sets of J, and thus $|c''((u, v)) \cap c''((u, v'))| \leq i$.
It remains to be shown that c'' is optimal. We have clearly $\chi_k^i(G\Box K_j) \geq max\{\chi_k^i(G), \chi_k^i(K_j)\}$. Since $\chi_k^i(G) \leq \chi_k^i(K_j)$ and since c'' uses $\chi_k^i(K_j)$ colors, it is indeed optimal and the proof is completed. □

We show now an analogous result for the product of two cycles using a similar proof technique.

Proposition 2.3.3. *Let C and D be two cycles, and let $|C| = r$, $|D| = s$. Then $\chi_k^i(C\Box D) = max\{\chi_k^i(C), \chi_k^i(D)\}$.*

Proof. Let $V(C) = v_0, \ldots, v_{r-1}$ and $V(D) = w_0, \ldots, w_{s-1}$. We start numbering colors and vertex labels from 0, as before. We will analyze three subcases, depending on the parity of r and s. In all cases, we will construct a k, i-coloring with $max\{\chi_k^i(C), \chi_k^i(D)\}$. Since $\chi_k^i(C\Box D) \geq max\{\chi_k^i(C), \chi_k^i(D)\}$, this is enough to prove the proposition.
Case 1: Both r and s are even. Then C and D are bipartite and this case is solved by Proposition 2.3.1.
Case 2: Both r and s are odd. Without loss of generality, let $r \geq s$. This means $\chi_k^i(C) \leq \chi_k^i(D)$, by Theorem 2.3.6. Let c be a k, i-coloring of D with $\chi_k^i(D)$ colors. Let F_s be the same family of bijections as in the previous proof, now on the set $\{0, 1, \ldots, s-1\}$. For each vertex v of each C-fiber C_w, define the following k, i-coloring c':

$$c'((v, w)) = \begin{cases} c(f_w(v)) & \text{if } v < s \\ c'((s-1, w)) & \text{if } v \geq s \text{ and } v \text{ odd} \\ c'((s, w)) & \text{if } v \geq s \text{ and } v \text{ even} \end{cases}$$

In words, the coloring c' assigns to the first s vertices of each C-fiber the color sets defined by c, successively rotating these color sets by one position for each fiber. For the remaining $r - s$ vertices in each fiber, assign the color set assigned to the s and $s - 1$ vertex of that fiber.
We have to show now that c' is a valid k, i-coloring of $C\Box D$. If this is the case, then it is optimal, since it employs $\chi_k^i(D)$ colors. Let $(v, w), (v', w) \in V(C\Box D)$, (v, w) adjacent

to (v', w). In this case, the two vertices belong to the same C-fiber C_w. By the proof of Lemma 2.3.4, c' is a valid k, i-coloring when restricted to a C-fiber, so this case is settled.
Let now $(v, w), (v, w') \in V(C \Box D)$, (v, w) adjacent to (v, w'). Thus (v, w) belongs to the C_w-fiber and (v, w') to the $C_{w'}$-fiber. We only need to analyze the case $v < s$, because for $v \geq s$ we merely repeat the last two values. Note that since $w, w' \in E(D)$, we may assume without loss of generality that $w' \equiv (w + 1) \pmod{s}$. We have hence:

$$\begin{aligned} w' &\equiv (w+1) \pmod{s} \\ (w'+v) &\equiv (w+v+1) \pmod{s} \\ (w'+v) \pmod{s} &\equiv (w+v+1) \pmod{s} \\ f_{w'}(v) &\equiv (w+v+1) \pmod{s} \end{aligned}$$

On the other hand, by the definition of modular addition, we have

$$\begin{aligned} (w+v+1) \pmod{s} &= (1 \pmod{s} + (w+v) \pmod{s}) \pmod{s} \\ &= (1 + f_w(v)) \pmod{s} \end{aligned}$$

So we have finally that $f_{w'}(v) \equiv (1 + f_w(v)) \pmod{s}$. This means that $c'((v, w))$ and $c((v, w'))$ are assigned color sets of c that correspond to (circularly) consecutive vertex labels in D, and hence to adjacent vertices. Thus it holds $|c'((v, w)) \cap c((v, w'))| \leq i$, and this case is finished.
Case 3: r is odd and s is even. Hence it holds that $\chi_k^i(C) \geq \chi_k^i(D)$. Color first the C-fiber C_0 with an optimal coloring c. Now color the fiber C_1 with a k, i-coloring c'' defined by $c''((v, 1)) = c((v+1) \bmod s)$. It is easy to see that c'' is a valid k, i-coloring of C_1. Finally, construct coloring c' for each fiber C_j as follows: color C_j like fiber C_0 if j is even and like fiber C_1 if j is odd. As c and c'' are valid colorings for any fiber C_j, it only remains to be shown that two adjacent vertices (v, u) and (v, w) belonging to different fibers C_u and C_w intersect in less than i colors. Since $uw \in E(D)$ and r is even, u and w must have different parity. Assume that u even, without loss of generality. Then $c'((v, u)) = c(v)$ and $c'((v, w)) = c((v + 1) \bmod s)$. As before, (circularly) correlative indices j and $j+1$ mean that the color sets c_j and c_{j+1} are assigned to adjacent vertices in the coloring c, so we have immediately $|c'((v, u)) \cap c'((v, w))| \leq i$. This case is also solved, since coloring c' uses only $\chi_k^i(C)$ colors. □

2.4 k, i-coloring of cliques

Brigham and Dutton proved the next partial results on the $(k : i)$-coloring of cliques:

Theorem 2.4.1. *[18]*
(a) If $n \leq \frac{k}{i} + 1$ then $\chi_k^{(i)}(K_n) = kn - \frac{n(n-1)i}{2}$.
(b) If $n \geq k^2 - k + 2$ then $\chi_k^{(i)}(K_n) = kn - (n-1)i$.

Part (a) of Theorem 2.4.1 also holds for $\chi_k^i(K_n)$. This was proved by Méndez-Díaz and Zabala in [83]. Part (b), however, does not. For a counterexample, let $n = 4$, $k = 2$ and

$i = 1$. We have that $\chi_2^{(1)}(K_4) = 5$, but $\chi_2^1(K_4) = 4$. Indeed, by Theorem 2.4.1 part (b), we have that $\chi_2^{(1)}(K_4) = 5$ and $\{\{1,2\},\{1,3\},\{1,4\},\{1,5\}\}$ is a proper $(2:1)$ coloring of K_4. On the other hand, $\{\{1,2\},\{1,3\},\{1,4\},\{3,4\}\}$ is a proper $2,1$-coloring of K_4, and thus $\chi_2^1(K_4) \leq 4$. By Proposition 2.4.1 below, we obtain that $\chi_2^1(K_4) \geq 4$.

The general problem of k, i-coloring cliques is still open, and it is also closely related to one of the central concerns in coding theory. We give now some definitions we need to present this relation. A *binary code* (or just a *code*, for brevity) is a set of binary vectors (or *codewords*) of length j. If a position in a binary vector contains a one, it will be called a *1-position* and a *0-position* otherwise. The *size* of a code is its cardinality. The *Hamming distance* of two codewords a and b is the number of positions in which they differ. The *distance* d_C of a code C is the smallest Hamming distance between any two codewords of C. A *(j, d, k)-constant weight code* is a set of codewords of length j and exactly k ones in each of them, with Hamming distance at least equal to d.
Given j, d and k, the question of determining the largest possible size $A(j, d, k)$ of a (j, d, k)-constant weight code has been studied for almost forty years, and remains one of the most basic questions in coding theory. The general answer is not known, but several upper and lower bounds on $A(j, d, k)$ have been found (see [1, 40] and references therein). We study now the relation between $A(j, d, k)$ and the k, i-coloring of cliques in the following Theorem:

Theorem 2.4.2. *A k, i-coloring for K_n with j colors does exist if and only if $A(j, 2(k-i), k) \geq n$.*

Proof. We start with the proof of necessity. Let f be a k, i-coloring of K_n with j colors. Construct a set $B = \{b_1, b_2, \ldots, b_n\}$ of n binary vectors, each of length j, such that every vector is the characteristic function of the set of colors associated with each vertex of K_n. That is, for every vertex v_s of K_n we have vector $b_s = (b_s^1, b_s^2, \ldots, b_s^j)$, where $b_s^t = 1$ if and only if color t belongs to $f(v_s)$. We will show that $d_B \geq 2(k-i)$. Let v_x and v_y be any two vertices of K_n, and b_x and b_y their associated binary vectors in B. Since $|f(v_x) \cap f(v_y)| \leq i$, b_x and b_y have at most i 1-positions in common. Vector b_x has k 1-positions in total, so at least $(k-i)$ 1-positions of b_x must be distributed along positions where b_y holds a 0. Analogously, vector b_y must also accommodate at least $(k-i)$ 1's along positions that store a 0 in b_x. This means that they differ in at least $2(k-i)$ positions, so $d(b_x, b_y) \geq 2(k-i)$. Since v_x and v_y are two arbitrary vertices of K_n, we have by definition of distance that $d_B \geq 2(k-i)$, so $A(j, 2(k-i), k) \geq n$.
We prove now sufficiency. Suppose $A(j, 2(k-i), k) \geq n$. Let B be a code that realizes $A(j, 2(k-i), k)$. Choose any n-subset of B. We have now only to interpret each binary vector $b \in B$ as a color set S_b, where a color c belongs to S_b if and only if $b_c = 1$. We obtain n color sets, each of cardinality k. By the same argument as before, no two of them have more than i colors in common, otherwise their corresponding binary vectors would be at a distance smaller than $2(k-i)$. Assign each set to a vertex of K_n. This is a valid k, i-coloring f that uses no more than j colors. □

By Theorem 2.4.2, we can rephrase the definition of the k, i-chromatic number of a complete graph K_n as the minimum positive integer j such that $A(j, 2(k-i), k) \geq n$.

This fact is used in the following straightforward corollary.

Corollary 2.4.2.1. *If $A(j, 2(k-i), k) \leq n$ and $m > n$, then $\chi_k^i(K_m) > j$.*

Thanks to Corollary 2.4.2.1, any upper bound on $A(j, d, k)$ for an even number d, can be used for generating new lower bounds for the $(k, k - \frac{d}{2})$-chromatic number of complete graphs. We will do so with the well known Johnson bound, presented in the next theorem:

Theorem 2.4.3. *[67] $A(j, 2r, k) \leq \lfloor \frac{rj}{k^2 - kj + rj} \rfloor$, if the denominator is positive.*

Let j be an integer such that $\frac{k^2}{i} > j$ (1). By Theorem 2.4.3 applied to $A(j, 2(k-i), k)$, we have that $A(j, 2(k-i), k) \leq \lfloor \frac{(k-i)j}{k^2 - ij} \rfloor$. Note that by our choice of j, the denominator is a positive number. Corollary 2.4.2.1 applied on this bound yields $\chi_k^i(K_n) > j$, if $n > \lfloor \frac{(k-i)j}{k^2 - ij} \rfloor$ (2). We are interested in the largest possible lower bound on $\chi_k^i(K_n)$, so we will find the maximum value for j that meets the given inequalities (1) and (2). For (2), we may write:

$$\begin{aligned} n &> \lfloor \frac{(k-i)j}{k^2 - ij} \rfloor \\ n &> \frac{(k-i)j}{k^2 - ij} \qquad (\text{If } x \in \mathbb{R}, n \in \mathbb{N}, n > x \iff n > \lfloor x \rfloor) \\ nk^2 &> (k-i)j + nij \\ \frac{nk^2}{(n-1)i + k} &> j \end{aligned}$$

For any real number x and any natural number j, we have $x > j \iff \lceil x \rceil > j$, so the largest possible value for j is $\lceil \frac{nk^2}{(n-1)i+k} \rceil - 1$. We show now that this value of j also meets (1):

$$\begin{aligned} \lceil \frac{nk^2}{(n-1)i + k} \rceil - 1 &\leq \lceil \frac{nk^2}{(n-1)i + i} \rceil - 1 \qquad (\text{Because } k \geq i) \\ &= \lceil \frac{k^2}{i} \rceil - 1 < \frac{k^2}{i} \end{aligned}$$

The second line holds since for all $x \in \mathbb{R}$, $\lceil x \rceil - x < 1$.
We have thus calculated our maximum possible j. Replacing this value of j in $\chi_k^i(K_n) > j$ gives rise to the following new lower bound on $\chi_k^i(K_n)$:

Proposition 2.4.1. $\chi_k^i(K_n) > \lceil \frac{nk^2}{(n-1)i+k} \rceil - 1$

We may as well take advantage of results on specific values of $A(j, d, k)$ found in the literature for achieving bounds on $\chi_k^i(K_n)$, for some values of n, k and i. We choose as an example a theorem due to Hanani:

Theorem 2.4.4. *[43, 44, 45, 46]*
(a) $A(j,6,4) = \frac{j(j-1)}{12}$*, if and only if* $j \equiv 1$ *or 4 (mod 12).*
(b) $A(j,8,5) = \frac{j(j-1)}{20}$*, if and only if* $j \equiv 1$ *or 5 (mod 20).*

Proposition 2.4.2. *Let* $j \equiv 1$ *or 4 (mod 12). Then*
(a) $\chi_4^1(K_n) > j$*, if* $n > \frac{j(j-1)}{12}$.
(b) $\chi_4^1(K_n) \leq j$*, if* $n \leq \frac{j(j-1)}{12}$.

Proof. Part (a) is a direct consequence of Theorem 2.4.4 (a) and Corollary 2.4.2.1. Part (b) follows from Theorem 2.4.4 (a) and Theorem 2.4.2. □

Proposition 2.4.3. *Let* $j \equiv 1$ *or 5 (mod 20). Then*
(a) $\chi_5^1(K_n) > j$*, if* $n > \frac{j(j-1)}{20}$.
(b) $\chi_5^1(K_n) \leq j$*, if* $n \leq \frac{j(j-1)}{20}$.

Proof. The proof is analogous to Proposition 2.4.2, using now Part (b) of Theorem 2.4.4. □

2.5 Resumen del capítulo

Un *coloreo por k-uplas* de un grafo G es una asignación de k colores a cada vértice de manera tal que vértices adyacentes tengan asignados colores diferentes. El mínimo número de colores necesario para un coloreo por k-uplas de G será denotado $\chi_k(G)$. Este problema fue planteado de manera independiente por varios autores. Hilton, Rado y Scott [50] utilizaron este coloreo como un concepto auxiliar para estudiar el denominado número cromático fraccionario en grafos planares. Stahl [92] estudió propiedades generales para coloreos por k-uplas y determinó χ_k para grafos bipartitos, grafos n-partitos completos y ciclos. Bollobás y Thomason [13] mostraron nuevas relaciones entre $\chi_k(G)$ y $\chi(G)$, en particular que $min\{\chi_k(G) : \chi(G) = j\} = 2k + j - 2$ y que $min\{\chi_k(G) : G \text{ tiene un único j-coloreo}\} = 2k + j - 1$.

Brigham y Dutton [18] generalizaron el concepto de coloreos por k-uplas al introducir el concepto de *$(k : i)$-coloreo*, donde los conjuntos de colores asignados a vértices adyacentes deben intersecar en exactamente i colores. El problema de $(k : i)$-coloreo consiste en encontrar el mínimo número de colores en un $(k : i)$-coloreo de un grafo G, que denotamos $\chi_k^{(i)}(G)$. Notar que $\chi_k^{(0)}(G) = \chi_k(G)$. En este trabajo, los autores determinaron $\chi_k^{(i)}(G)$ para grafos bipartitos y ciclos impares. Se obtuvieron también resultados parciales para grafos completos.

Otra generalización, conocida como *k, i-coloreo*, fue introducida por Méndez-Díaz y Zabala en [83], en la cual los conjuntos de colores pueden intersecar en a lo sumo i colores. Formalmente, sea G un grafo y sean k, i, j números no negativos, con $0 \leq i \leq k \leq j$. Entonces un k, i-coloreo de G con j colores consiste en asignar a cada vértice v de G un conjunto $c(v) \subseteq \{1, \ldots, j\}$ de cardinalidad k tal que cada par de vértices adyacentes u, v verifica $|c(v) \cap c(u)| \leq i$. El mínimo número positivo j tal que G admite un k, i-coloreo con j colores es llamado *número k, i-cromático* y es denotado $\chi_k^i(G)$. Notar que para $k = 1$, $i = 0$, tenemos el coloreo clásico, entonces $\chi_1^0(G) = \chi(G)$ para cualquier grafo G. Para k arbitrario e $i = 0$, tenemos el coloreo por k-uplas.

Méndez-Díaz y Zabala resolvieron en [83] el problema de k, i-coloreo para algunos valores de k e i en grafos completos, estudiaron la noción de perfección y criticidad para el problema de k, i-coloreo y dieron cotas generales para el número k, i-cromático. Las autoras propusieron además un enfoque heurístico y un modelo de programación lineal para el problema, que continuaron desarrollando y generalizando en [84].

Notar que $\chi_k^i(G) \leq \chi_k^{(i)}(G)$, ya que cada $(k : i)$-coloreo es en particular un k, i-coloreo, pero no son necesariamente iguales, aún para grafos completos.

En este capítulo, damos las siguientes nuevas cotas generales para χ_k^i:

Proposición: $\chi_k^i(G) < \chi_{k+1}^i(G)$.

Proposición: Sea G un grafo con por lo menos una arista, y sea $k < i$. Entonces $\chi_k^{i+1}(G) < \chi_k^i(G)$.

Presentamos además un algoritmo lineal para computar el número k, i-cromático de los ciclos:

Teorema: Sea C_n un ciclo, con $n = 2t + 1$. Entonces $\chi_k^i(C_n) = max\{2k - i, \lceil \frac{n(k-i)}{t} \rceil\}$ y se puede obtener un k, i-coloreo de C_n con $\chi_k^i(C_n)$ colores en tiempo $O(n)$.

Un grafo G es un *cactus* si no contiene dos ciclos que compartan una arista.

Teorema: Sea G un cactus. Entonces, un k, i-coloreo de G con $\chi_k^i(G)$ puede ser computado en tiempo lineal.

Mostramos además que estos resultados valen para el número $(k : i)$-cromático de ciclos y cactus.

Estudiamos luego el número k, i-cromático de algunos productos cartesianos:

Proposición: Si $\chi(G) = j$, entonces $\chi_k^i(G \square K_j) = max\{\chi_k^i(G), \chi_k^i(K_j)\}$.

Proposición: Sean C y D dos ciclos, y sea $|C| = r$, $|D| = s$. Entonces $\chi_k^i(C \square D) = max\{\chi_k^i(C), \chi_k^i(D)\}$.

Finalmente, presentamos una relación entre el problema de k, i-coloreo en grafos completos y códigos binarios pesados. Un *código de peso constante* (j, d, k) es un conjunto C de vectores binarios de longitud j, con exactamente k unos en cada vector, tal que dos vectores cualquiera difieran en al menos d posiciones. Dados j, d and k, determinar la cardinalidad máxima de un código de peso constante (j, d, k) es uno de los problemas abiertos más importantes en teoría de códigos. Denotamos una instancia de este problema $A(j, d, k)$.

Teorema: Existe un k, i-coloreo para K_n con j colores si y sólo si $A(j, 2(k-i), k) \geq n$.

Los resultados en este capítulo fueron enviados para su publicación en [15].

CHAPTER 3

On the b-coloring of P_4-tidy graphs

A *b-coloring* of a graph is a coloring such that every color class must have a vertex adjacent to at least one vertex of every other color class. The *b-chromatic number* of a graph G, denoted by $\chi_b(G)$, is the maximum number t such that G has a b-coloring with t colors. The b-coloring parameter was introduced by R. W. Irving and D. F. Manlove [56] by considering proper colorings that are minimal with respect to a partial order P defined on the set of all partitions of the vertex set of G. The authors proved in that work that determining the b-chromatic number of a graph G is NP-hard, but polynomially solvable for trees. Kratochvíl, Tuza and Voigt showed in [77] that determining $\chi_b(G)$ is NP-hard even if G is a connected bipartite graph, but can be solved in polynomial time for some families of bipartite graphs. Bonomo et al. also proved that the problem remains NP-hard on co-bipartite graphs, but polynomially solvable on tree-cographs [17]. Corteel et al. [30] showed that the problem is also hard to approximate in polynomial time within a factor of $\frac{120}{113} - \epsilon$, for any $\epsilon > 0$ unless $P = NP$.

Since the seminal paper by Irving and Manlove, much effort has been devoted to solving the problem for specific graph classes, both deriving exact solutions and bounds on χ_b. The b-chromatic number of power graphs of complete caterpillars was studied in [32], of power graphs of paths and power graphs of cycles in [33], and of power graphs of complete k-ary trees in [34]. Javadi and Omoomi [64] determined $\chi_b(G)$ for Kneser graphs $K(n,k)$ for some values of n and k. It was proven in [57] that the b-chromatic number of cubic graphs is 4, except for four particular graphs (the Petersen graph, $K_{3,3}$, the prism over $K_{3,3}$ and a 10 vertex graph). The same authors solved $\chi_b(G)$ for cartesian products of paths and cycles with complete graphs and cartesian product of two complete graphs in [65]. Kouider and Mahéo present in [75] lower bounds for the b-chromatic number of the cartesian product of two graphs. Jakovac and Peterin

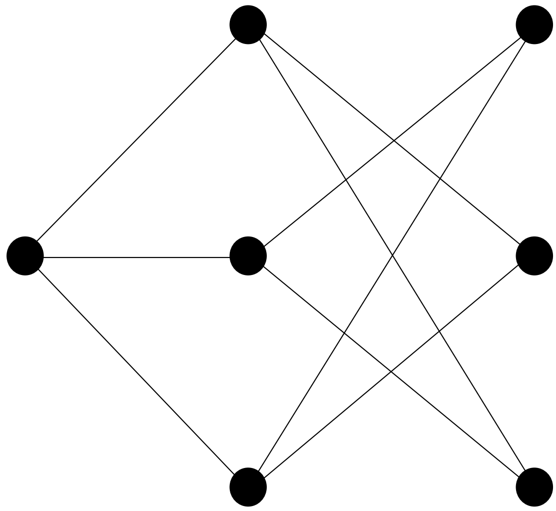


Figure 3.1: The smallest non b-continuous graph G (see [2]). Note that G has a b-coloring with 2 and with 4 colors, but none with 3 colors.

[59] determine some upper and lower bounds for the b-chromatic number of the strong product, the lexicographic product and the direct product. They also give some exact values for products of paths, cycles, stars, and complete bipartite graphs. The b-chromatic number of regular graphs were studied in [12, 35, 20]. Central graphs of some complete bipartite graphs, cycles and paths were analyzed in [93], as well as middle graphs of cycles, paths, wheel graphs and fan graphs in [96]. In [95] the central, middle and total graph of a star $K_{1,n}$ is studied. Campos, Linhares Sales, Maffray and Silva studied $\chi_b(G)$ for cacti [23], and Havet, Linhares Sales and Sampaio for the so called tight graphs [48]. In [22] Campos et al. showed how to compute in polynomial time the b-chromatic number of a graph of girth at least 9, generalizing a previous result from [81] on outerplanar graphs of large girth. Bounds for $\chi_b(G)$ for the Mycielskian of some families of graphs was studied in [7], for the graph $G - v$ in [6], and for generalized Hamming graphs in [25]. In turn, upper bounds for $K_{1,s}$-free graphs, graphs with given minimum clique partition and bipartite graphs were given in [76]. General upper bounds for the b-chromatic number were determined in [2].

The behavior of the b-chromatic number can be surprising. In contrast with classic coloring, the values of k for which a graph has a b-coloring with k colors do not necessarily form an interval of the set of integers; in fact any finite subset of $\mathbb{N}_{\geq 2}$ can constitute the set of these values for some graph [9]. A graph G is *b-continuous* if it has a b-coloring with t colors, for every $t = \chi(G), \ldots, \chi_b(G)$. In Figure 3.1 we show a non b-continuous graph. In [68] it is proved that chordal graphs and some planar graphs are b-continuous.

Another atypical property is that the b-chromatic number can increase when taking induced subgraphs. A graph G is defined to be *b-monotonic* if $\chi_b(H_1) \geq \chi_b(H_2)$ for every induced subgraph H_1 of G, and every induced subgraph H_2 of H_1 [16]. See Figure

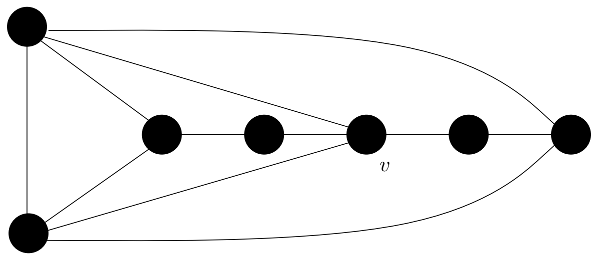


Figure 3.2: A non b-monotonic graph. Note that $\chi_b(G) = 3$, but removal of vertex v makes $\chi_b(G - v) = 4$

3.2 for an example of a non b-monotonic graph.

Recall that in Chapter 1 we motivated the definition for the b-chromatic number with an heuristic that started from a given coloring and tried to decrease the number of colors by eliminating color classes. But when does this heuristic actually produce an exact result for $\chi(G)$? Hoáng and Kouider introduced for this the notion of b-perfect graphs in [52]. A graph G is *b-perfect* if $\chi_b(H) = \chi(H)$ for every induced subgraph H of G, so b-perfect graphs are indeed optimally colored by the heuristic. b-perfect bipartite and P_4-sparse graphs were characterized in [52]. In the same article, it is also proved that every $2K_2$-free and $\overline{P_5}$-free graph is b-perfect. In [53], Hoàng, Linhares Sales and Maffray presented the following conjecture: A graph is b-perfect if and only if does not contain any one of twenty-two forbidden subgraphs. The conjecture was shown to be true for diamond free graphs, 3-colorable graphs [53] and some chordal graphs [80]. Recently, Maffray, Hoáng and Mechebbek proved it correct for general graphs in [54], leading to polynomial recognition of b-perfect graphs.

A *cograph* is a graph that does not contain P_4 as an induced subgraph. This class of graphs has been defined independently by many authors. Corneil et al. proposed a linear recognition algorithm based on a unique decomposition of the cograph [27]. Several generalizations of cographs have been defined in the literature, such as P_4-sparse [51], P_4-lite [60], P_4-extendible [62] and P_4-reducible graphs [61]. A graph class generalizing all of them is the class of P_4-tidy graphs [38]. Let G be a graph and A a P_4 in G. A *partner* of A is a vertex v in $G - A$ such that $A \cup \{v\}$ induces at least two P_4s in G. A graph G is *P_4-sparse* if no induced P_4 has a partner and *P_4-tidy* if every induced P_4 has at most one partner (see Figure 3.3 for an example graph). Another generalization of P_4-sparse graphs are $(q, q-4)$-graphs. A graph is a *$(q, q-4)$-graph* if no set of at most q vertices induces more than $q - 4$ distinct P_4's [4]. In [24] Campos, Linhares Sales, Maia and Sampaio obtain a polynomial time algorithm for the b-chromatic number of $(q, q - 4)$-graphs, for a fixed q. There is no containment relationship between the classes P_4-tidy and $(q, q - 4)$-graph.

We prove in this chapter that P_4-tidy graphs are b-continuous and b-monotonic. Furthermore, we describe a polynomial time algorithm to compute the b-chromatic number for this class of graphs. We extend thus the results presented in [16] for the class of P_4-sparse graphs. The results given in this chapter were published in [11].

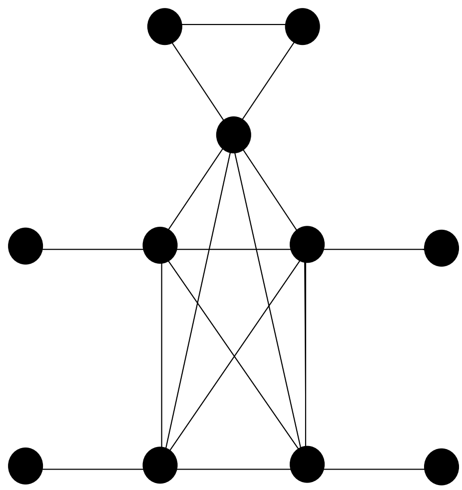


Figure 3.3: A P_4-tidy graph

3.1 Definitions and preliminary results

Two vertices will be said to be *true twins* if they are adjacent and have the same neighborhood, and *false twins* if they are non-adjacent but have the same neighbors. A vertex is *simplicial* if its neighbors induce a complete subgraph. A vertex v *controls* a vertex w if v and w are non-adjacent and all the neighbors of w are neighbors of v.

Lemma 3.1.1. [53] *Let G be a graph and φ a coloring of G. If v and w are false twins in G, then either none of them is dominant, or $\varphi(v) = \varphi(w)$.*

This can be extended straightforwardly to the following one.

Lemma 3.1.2. *Let G be a graph and φ a coloring of G. If v controls w, then if w is dominant, so is v and $\varphi(v) = \varphi(w)$.*

Lemma 3.1.3. [53] *Let G be a graph and φ a coloring of G with more than $\chi(G)$ colors. Then no simplicial vertex of G is dominant.*

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. The *union* of G_1 and G_2 is the graph $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$. The union is clearly an associative operation and, for each nonnegative integer t, we will denote by tG the union of t disjoint copies of G. The *join* of G_1 and G_2 is the graph $G_1 \vee G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$. That is, the vertex set of $G_1 \vee G_2$ is $V_1 \cup V_2$ and its edge set is $E_1 \cup E_2$ plus all the possible edges with an endpoint in V_1 and the other one in V_2.

Cographs can be built from isolated vertices by using these two operations.

Theorem 3.1.4. [27] *Every non-trivial cograph is either union or join of two smaller cographs.*

Thus, the chromatic number of a cograph can be recursively calculated due to the following result.

Theorem 3.1.5. [28] *If G is the trivial graph, then $\chi(G) = 1$. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. Then,*

i. $\chi(G_1 \cup G_2) = max\{\chi(G_1), \chi(G_2)\}$
ii. $\chi(G_1 \vee G_2) = \chi(G_1) + \chi(G_2)$.

A similar result holds for the b-chromatic number, but the relation between the b-chromatic number of two graphs and the b-chromatic number of their union is weaker.

Theorem 3.1.6. [73] *If G is the trivial graph, then $\chi_b(G) = 1$. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. Then,*

i. $\chi_b(G_1 \cup G_2) \geq max\{\chi_b(G_1), \chi_b(G_2)\}$
ii. $\chi_b(G_1 \vee G_2) = \chi_b(G_1) + \chi_b(G_2)$.

P_4-tidy graphs have also a useful decomposition theorem. We will use it extensively in this work to inductively prove our results. A brief description of the theorem follows.

Let $G = (V, E)$ be a graph. Let $F = \{e \in E \mid e$ belongs to an induced P_4 of $G\}$. Let $G_p = (V, F)$. A connected component of G_p having exactly one vertex is called a *weak vertex*. Any connected component of G_p distinct from a weak vertex is called a *p-component* of G. A graph G is *p-connected* if it has only one p-component and no weak vertices [5].

A p-connected graph $G = (V, E)$ is *p-separable* if V can be partitioned into two sets (C, S) such that each P_4 that contains vertices from C and from S has its midpoints in C and its endpoints in S. We will call it a *p-partition*. If such a partition exists, then it is unique [63].

An *urchin* (resp. *starfish*) of size k, $k \geq 2$, is a p-separable graph with p-partition (C, S), where $C = \{c_1, \ldots, c_k\}$ is a clique; $S = \{s_1, \ldots, s_k\}$ is a stable set; s_i is adjacent to c_i if and only if $i = j$ (resp. $i \neq j$).

A *quasi-urchin* (resp. *quasi-starfish*) of size k is a graph obtained from an urchin (resp. starfish) of size k by replacing at most one vertex by K_2 or S_2. Note that the new vertices result on true or false twins, respectively, and they are in the same set of the new p-partition (C^*, S^*). The elements of S^* are called the *legs* and C^* is called the *body* of the quasi-starfish or quasi-urchin.

Note that there are five possible quasi-starfishes of size two, and they are also the five possible quasi-urchins of size two: P_4, P, $\overline{P}$, fork and kite (see Figure 3.4). To avoid ambiguity, we will consider these five graphs as quasi-starfishes, while quasi-urchins will be always of size at least three.

When considering quasi-urchins and quasi-starfishes, we have ten kinds of them. We will call *type 1* (resp. *type 2*) the urchins (resp. starfishes); *type 3* (resp. *type 4*) the urchins (resp. starfishes), where a vertex in the body was replaced by K_2; *type 5* (resp.

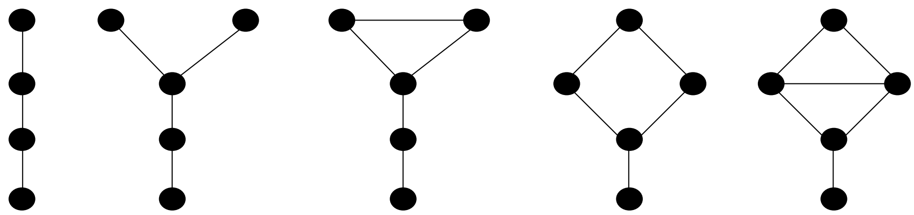


Figure 3.4: Possible quasi-starfishes of size two. From left to right: P_4, fork, $\overline{P}$, P and kite.

type 6) the urchins (resp. starfishes), where a vertex in the body was replaced by S_2; *type 7* (resp. *type 8*) the urchins (resp. starfishes), where a leg was replaced by K_2; and *type 9* (resp. *type 10*) the urchins (resp. starfishes), where a leg was replaced by S_2. Recall that graphs of odd type have always size at least three and, with this condition, the ten types form a partition over the family of quasi-urchins and quasi-starfishes.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \emptyset$, such that G_1 is p-separable with partition (V_1^1, V_1^2). Consider the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2 \cup \{xy \mid x \in V_1^1, y \in V_2\}$. We shall denote this graph by $G_1 \,\underline{\vee}\, G_2$.

Theorem 3.1.7. [63] *Every graph G either is p-connected or can be obtained uniquely from its p-components and weak vertices by a finite sequence of $\cup$, $\vee$ and $\underline{\vee}$ operations.*

Proposition 3.1.1. [38]*A graph G is P_4-tidy if and only if every p-component is isomorphic to either P_5 or $\overline{P_5}$ or C_5 or a quasi-starfish or a quasi-urchin. Quasi-starfishes and quasi-urchins are the p-separable p-components of G.*

Remark 3.1.1. *Let G_1 be a quasi-urchin or a quasi-starfish, and G_2 be a graph. If G_1 is type 7 or 8, all the legs are simplicial vertices both in G_1 and in $G_1 \,\underline{\vee}\, G_2$. Otherwise, both in G_1 and in $G_1 \,\underline{\vee}\, G_2$, each leg of G_1 is controlled by a vertex in the body of G_1. Then, by Lemmas 3.1.2 and 3.1.3, for every coloring of G_1 (resp. $G_1 \,\underline{\vee}\, G_2$) with more than $\chi(G_1)$ (resp. $\chi(G_1 \underline{\vee} G_2)$) colors, if there is a dominant vertex of color c in $V(G_1)$, then there is a dominant vertex of color c in the body of G_1.*

Lemma 3.1.8. *Let G be a quasi-starfish or quasi-urchin of size k. Then,*

i. If G is type 1,2,5,6,7,9 or 10, then $\chi(G) = k$.
ii. if G is type 3,4 or 8, then $\chi(G) = k + 1$.
iii. $\chi_b(G) = \chi(G)$.

Proof. Items *i.* and *ii.* are easy to prove, since a coloring of the maximum clique of G can be extended to the whole graph without increasing the number of colors. Let (C^*, S^*) be the p-partition of G. To prove item *iii.*, suppose on the contrary that we have a b-coloring φ of G with more than $\chi(G)$ colors. By Remark 3.1.1, if there is a dominant vertex of color c in G, then there is a dominant vertex of color c in C^*. If G is neither type 5 nor type 6, then $\chi_b(G) \leq |C^*| \leq \chi(G)$, a contradiction. If G is type 5 or 6, then there is a pair of false twins in C^*, so by Lemma 3.1.1, at most

$|C^*| - 1$ different colors can have dominant vertices and $\chi_b(G) \leq |C^*| - 1 \leq \chi(G)$, a contradiction. □

Lemma 3.1.9. *Let $G_1 = (V_1, E_1)$ be a p-separable P_4-tidy graph, and $G_2 = (V_2, E_2)$ a graph such that $V_1 \cap V_2 = \emptyset$. Then,*

i. If G_1 is not type 8, then $\chi(G_1 \veebar G_2) = \chi(G_1) + \chi(G_2)$; if G_1 is type 8, then $\chi(G_1 \veebar G_2) = \chi(G_1) + \chi(G_2) - 1$.
ii. If G_1 is not type 8, then $\chi_b(G_1 \veebar G_2) = \chi_b(G_1) + \chi_b(G_2)$; if G_1 is type 8, then $\chi_b(G_1 \veebar G_2) = \chi_b(G_1) + \chi_b(G_2) - 1$.

Proof. Let $G = G_1 \veebar G_2$. By Proposition 3.1.1, G_1 is a quasi-urchin or a quasi-starfish. Let (C^*, S^*) be its p-partition. Then G contains $G_1[C^*] \vee G_2$ as an induced subgraph, thus $\chi(G) \geq \chi(G_1[C^*] \vee G_2)$. On the other hand, every coloring of $G_1[C^*] \vee G_2$ can be extended to G without adding new colors, by giving to each vertex in S^* either a color used by a non-neighbor of it in C^* or a color used in G_2. Hence, $\chi(G) = \chi(G_1[C^*] \vee G_2)$. By Theorem 3.1.5, $\chi(G_1[C^*] \vee G_2) = \chi(G_1[C^*]) + \chi(G_2)$. By Lemma 3.1.8, if G_1 is type 8 then $\chi(G_1[C^*]) = \chi(G_1) - 1$, otherwise $\chi(G_1[C^*]) = \chi(G_1)$. This concludes the proof of item *i.*.

In order to prove item *ii.*, we will show that $\chi_b(G) = \chi_b(G_2) + \chi(G_1[C^*])$. Any b-coloring of G_2 can be extended to a b-coloring of G by assigning $\chi(G_1[C^*])$ new colors to C^* and giving to each vertex in S^* either a color used by a non-neighbor of it in C^* or a color used in G_2. So, $\chi_b(G) \geq \chi_b(G_2) + \chi(G_1[C^*])$.

If $\chi_b(G) = \chi(G)$, by item *i.*, $\chi_b(G) = \chi(G_2) + \chi(G_1[C^*]) \leq \chi_b(G_2) + \chi(G_1[C^*])$. So, we may suppose $\chi_b(G) > \chi(G)$. Let now φ be a b-coloring of G with more than $\chi(G)$ colors. By Remark 3.1.1, if there is a dominant vertex of color c in G, then there is a dominant vertex of color c in $C^* \cup V(G_2)$. Notice that the set of colors used by vertices in G_2 and the set of colors used in C^* are disjoint, so C^* should contain dominant vertices for all the colors used in $V(C^*)$. In particular, if G_1 is type 5 or 6, by Lemma 3.1.1, it follows that the two non-adjacent vertices in C^* receive the same color, thus C^* is colored with $\chi(G_1[C^*])$ colors. On the other hand, it is easy to see that φ restricted to $V(G_2)$ is a b-coloring of G_2. So $\chi_b(G) \leq \chi_b(G_2) + \chi(G_1[C^*])$.

We have proved that $\chi_b(G) = \chi_b(G_2) + \chi(G_1[C^*])$. By Lemma 3.1.8, if G is type 8 then $\chi(G_1[C^*]) = \chi(G_1) - 1 = \chi_b(G_1) - 1$, otherwise $\chi(G_1[C^*]) = \chi(G_1) = \chi_b(G_1)$. This concludes the proof of item *ii.*. □

3.2 b-continuity in P_4-tidy graphs

In [16], a family of cographs with arbitrarily large difference between their b-chromatic number and their chromatic number was shown. Therefore, it makes sense to analyze b-continuity in P_4-tidy graphs. In this section we prove that P_4-tidy graphs are b-continuous, by using the decomposition theorem for this class of graphs.

Lemma 3.2.1. *If G is P_5, $\overline{P_5}$, C_5, a quasi-urchin or a quasi-starfish, then G is b-continuous.*

Proof. If $G = P_5$, then $\chi(G) = 2$ and $\chi_b(G) = 3$. If $G = \overline{P_5}$ or $G = C_5$, we have $\chi(G) = \chi_b(G) = 3$. Finally, quasi-starfishes and quasi-urchins are b-continuous by Lemma 3.1.8, item *iii*. □

Lemma 3.2.2. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. If G_1 and G_2 are b-continuous and $G = G_1 \cup G_2$, then G is b-continuous.*

Lemma 3.2.3. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. If G_1 and G_2 are b-continuous and $G = G_1 \vee G_2$, then G is b-continuous.*

Lemma 3.2.4. *Let $G_1 = (V_1, E_1)$ be a p-separable P_4-tidy graph and $G_2 = (V_2, E_2)$ be a graph such that $V_1 \cap V_2 = \emptyset$. If G_2 is b-continuous and $G = G_1 \veebar G_2$, then G is b-continuous.*

Proof. By Proposition 3.1.1, G_1 is a quasi-starfish or a quasi-urchin. Let (C^*, S^*) be the p-partition of G_1. Suppose first that G_1 is not type 8. Any b-coloring of G_2 with t colors $\{1, \ldots, t\}$ can be extended to a b-coloring of G with $t + \chi(G_1)$ colors, in the following way. If we color G_1 using colors $\{t+1, \ldots, t+\chi(G_1)\}$, then every dominant vertex in G_2 will have now also neighbors with colors $t+1, \ldots, t+\chi(G_1)$, and every dominant vertex in C^* will have now also neighbors with colors $1, \ldots, t$. Since C^* contains dominant vertices of all colors in $\{t+1, \ldots, t+\chi(G_1)\}$, the resulting coloring is a b-coloring of G with $t + \chi(G_1)$ colors.

Suppose now that G_1 is type 8. Any b-coloring of G_2 with t colors $\{1, \ldots, t\}$ can be extended to a b-coloring of G with $t + \chi(G_1) - 1$ colors, in the following way. If we color G_1 using colors $\{t, \ldots, t + \chi(G_1) - 1\}$ in such a way that C^* uses colors from $t+1$ to $t + \chi(G_1) - 1$, then every dominant vertex in G_2 will have now also neighbors with colors $t+1, \ldots, t+\chi(G_1)-1$, and every dominant vertex in C^* will have now also neighbors with colors $1, \ldots, t$. Since C^* contains dominant vertices of all colors in $\{t+1, \ldots, t+\chi(G_1)-1\}$, this results in a b-coloring of G with $t + \chi(G_1) - 1$ colors.

Since G_2 is b-continuous, we can obtain b-colorings for G with each color t', where $\chi(G_2) + \chi(G_1) \leq t' \leq \chi_b(G_2) + \chi_b(G_1)$ in the first case, and $\chi(G_2) + \chi(G_1) - 1 \leq t' \leq \chi_b(G_2) + \chi_b(G_1) - 1$ in the second case. By Lemma 3.1.9, $\chi(G) = \chi(G_2) + \chi(G_1)$ and $\chi_b(G) = \chi_b(G_2) + \chi_b(G_1)$ in the first case, while $\chi(G) = \chi(G_2) + \chi(G_1) - 1$ and $\chi_b(G) = \chi_b(G_2) + \chi_b(G_1) - 1$ in the second case, so G is b-continuous. □

Theorem 3.2.5. *P_4-tidy graphs are b-continuous.*

Proof. Immediate by an inductive argument using the decomposition Theorem 3.1.7, Proposition 3.1.1, Lemmas 3.2.2, 3.2.3 and 3.2.4 and Lemma 3.2.1 for the base case of the induction. □

3.3 Computation of the b-chromatic number in P_4-tidy graphs

The inequality in part *i.* of Theorem 3.1.6 can be strict, and this fact prevents us from using this result for directly computing the b-chromatic number of P_4-tidy graphs by using the decomposition Theorem 3.1.7. In fact, it is not difficult to build examples showing that the b-chromatic number of the graph $G_1 \cup G_2$ does not depend only on the b-chromatic numbers of G_1 and G_2. To overcome this problem, we follow the approach in [16] in the definition of the *dominance sequence* $dom_G \in \mathbb{Z}^{\mathbb{N}_{\geq \chi(G)}}$ of a graph G, where $dom[t]$ is the maximum number of distinct color classes that admit dominant vertices in any coloring of G with t colors, for $\chi(G) \leq t \leq |V(G)|$. We will compute this sequence recursively on P_4-tidy graphs by using the decomposition theorem. Then we will obtain the b-chromatic number of G as the maximum t such that $dom_G[t] = t$.

Lemma 3.3.1. *Let G be P_5, $\overline{P_5}$, C_5, a quasi-urchin or a quasi-starfish. The dominance sequence for G can be obtained in linear time.*

Proof. It is easy to see that $dom_{P_5}[2] = 2$, $dom_{P_5}[3] = 3$, and $dom_{P_5}[t] = 0$ for $t \geq 4$; $dom_{\overline{P_5}}[3] = 3$, $dom_{\overline{P_5}}[4] = 1$, and $dom_{\overline{P_5}}[5] = 0$; $dom_{C_5}[3] = 3$, and $dom_{C_5}[t] = 0$ for $t \geq 4$. Now, let $G = (C^*, S^*)$ be a quasi-urchin or quasi-starfish of size k. Let (C, S) be the p-partition of the urchin or starfish, $S = \{s_1, \ldots, s_k\}$, $C = \{c_1, \ldots, c_k\}$. If a vertex in S (resp. C) was replaced by two vertices, we will assume that the vertex was s_1 (resp. c_1) and that it was replaced by vertices s_1', s_1'' (resp. c_1', c_1''). Recall that, for every graph G, $dom_G[\chi(G)] = \chi(G)$. Consider now colorings of G with more than $\chi(G)$ colors. By Remark 3.1.1, if there is a dominant vertex of color c in G, then there is a dominant vertex of color c in C^*. So, for $t > \chi(G)$, $dom_G[t] \leq |C^*|$.

If G is type 1, then $dom_G[k] = dom_G[k+1] = k$ and $dom_G[t] = 0$ for $t \geq k+2$; if G is type 2, then $dom_G[k+s] = \min\{k, 2k-2s\}$ for $0 \leq s \leq k$, and $dom_G[t] = 0$ for $t > 2k$ [16].

We start by analyzing the different kinds of quasi-urchins.

Claim 1. If G is type 3, then $dom_G[k+1] = dom_G[k+2] = k+1$, $dom_G[t] = 0$ for $t \geq k+3$.

In G there are $k+1$ vertices of degree $k+1$ and no vertex of degree at least $k+2$, so the upper bounds for each value of dom_G are clear (a dominant vertex in a coloring with t colors must have degree at least $t-1$). A coloring with $k+2$ colors and $k+1$ dominant vertices of different colors can be obtained by coloring all the vertices in S^* with the same color, different from the colors used in C^*. $\diamondsuit$

Claim 2. If G is type 5, then $dom_G[k] = dom_G[k+1] = k$, $dom_G[k+2] = k-1$, $dom_G[t] = 0$ for $t \geq k+3$.

Since $k \geq 3$, in G there are $k-1$ vertices of degree $k+1$, 2 vertices of degree k, and no vertex of degree at least $k+2$. So, the upper bounds on $dom_G[t]$ for $t \geq k+2$ are clear. The upper bound for $dom_G[k+1]$ holds by Lemma 3.1.1. Two colorings attaining the upper bounds for $dom_G[k+1]$ and $dom_G[k+2]$ are defined as follows. Vertices

$c_2, \ldots, c_k$ receive colors $1, \ldots, k-1$; vertices $s_1, \ldots, s_k$ receive color $k+1$; vertices c'_1, c''_1 receive both color k or colors k and $k+2$, respectively. $\diamondsuit$

Claim 3. If G is type 7 or type 9, then $dom_G[k] = dom_G[k+1] = k$, $dom_G[k+2] = 1$, $dom_G[t] = 0$ for $t \geq k+3$.

Since $k \geq 3$, in G there are $k-1$ vertices of degree k, one vertex of degree $k+1$, and no vertex of degree at least $k+2$. So, the upper bounds on $dom_G[t]$ are clear. Two colorings attaining the upper bounds for $dom_G[k+1]$ and $dom_G[k+2]$ are defined as follows. Vertices $c_1, \ldots, c_k$ receive colors $1, \ldots, k$; vertices $s'_1, s_2, \ldots, s_k$ receive color $k+1$; vertex s''_1 receives color 2 or $k+2$, respectively. $\diamondsuit$

We will now analyze the different kinds of quasi-starfishes.

Claim 4. If G is type 4, then $dom_G[k+1+s] = \min\{k, 2k-2s\}+1$ for $0 \leq s < k$, and $dom_G[t] = 0$ for $t > 2k$.

Since $\chi(G) = k+1$, then $dom_G[k+1] = k+1$. Since the maximum degree of G is $2k-1$, it is clear that $dom_G[t] = 0$ for $t > 2k$. Let $t = k+1+s$ such that $1 \leq s \leq k-1$ and let φ be a coloring of G with t colors and maximum number of colors with dominant vertices. At least one of c'_1, c''_1 has a color different from $\varphi(s_1)$. Suppose without loss of generality that $\varphi(c''_1) \neq \varphi(s_1)$, then $\varphi(c''_1) \neq \varphi(v)$ for every $v \in V(G)$. Let $G' = G - \{c''_1\}$. Thus the restriction of φ to G' is a coloring with $t-1$ colors, and dominant vertices of G are still dominant in G', therefore $dom_G[t] \leq dom_{G'}[t-1]+1$. Conversely, let ψ be a coloring of G' with $t-1$ colors (namely, colors $1, \ldots, t-1$) and maximum number of colors with dominant vertices. We can extend ψ to a t-coloring of G by defining $\psi(c''_1) = t$. Since $t-1 \geq k+1$, no vertex in S^* was dominant in G', so every dominant vertex of G' is still dominant in G. Besides, c''_1 is now dominant in G if and only if $\psi(s_1) = \psi(v)$ for some vertex v of G', different from s_1, and this happens if and only if c'_1 was dominant in G'. By symmetry of G', we may assume that if $dom_{G'}[t-1] > 0$ then c'_1 was dominant in G'. So, if $dom_{G'}[t-1] > 0$, we have that $dom_G[t] = dom_{G'}[t-1]+1$. Since G' is type 2, we already know that $dom_{G'}[k+s] = \min\{k, 2k-2s\}$. Since $s \leq k-1$, $dom_{G'}[t-1] > 0$, and $dom_G[k+1+s] = \min\{k, 2k-2s\}+1$. $\diamondsuit$

Claim 5. If G is type 6, then $dom_G[k+s] = k$ for $0 \leq s \leq \lfloor \frac{k}{2} \rfloor$, $dom_G[k+s] = \min\{k-1, 2k-2s+2\}$ for $\lfloor \frac{k}{2} \rfloor \leq s \leq k$, and $dom_G[k+s] = 0$ for $s > k$.

Since $\chi(G) = k$, then $dom_G[k] = k$. Since the maximum degree of G is $2k-1$, it is clear that $dom_G[t] = 0$ for $t > 2k$. Let $t = k+s$ with $1 \leq s \leq k$ and let φ be a coloring of G with t colors and maximum number of colors with dominant vertices.

Suppose first that $\varphi(c'_1) = \varphi(c''_1)$. Then the number of colors with dominant vertices in G is the same as the number of colors with dominant vertices when restricting φ to $G' = G - \{c''_1\}$. Conversely, any coloring of G' can be extended to a coloring of G by giving to c''_1 the color used by c'_1, thus preserving the dominant vertices. Then, if $\varphi(c'_1) = \varphi(c''_1)$, it follows that $dom_G[k+s] = dom_{G'}[k+s]$ and, since G' is type 2, $dom_G[k+s] = \min\{k, 2k-2s\}$.

Suppose now that $\varphi(c'_1) \neq \varphi(c''_1)$. By Lemma 3.1.1, none of c'_1, c''_1 is dominant. So, in this case, the number of colors with dominant vertices is at most $k-1$. We may assume $2s > k$, otherwise, by the arguments above, we can find a coloring φ' of G with $\varphi'(c_1') = \varphi'(c_1'')$ and such that there are k colors with dominant vertices. Since $k \geq 2$, this implies $s > 1$, hence $t > k+1$. Since $\varphi(c_1') \neq \varphi(c_1'')$, at least one of them has a color different from $\varphi(s_1)$. Suppose without loss of generality that $\varphi(c_1'') \neq \varphi(s_1)$, then $\varphi(c_1'') \neq \varphi(v)$ for every $v \in V(G)$. Let $G' = G - \{c_1''\}$. Thus the restriction of φ to G' is a coloring with $t-1$ colors, and dominant vertices of G are still dominant in G'. Since c_1'' was not dominant in G, the number of colors with dominant vertices does not decrease. Conversely, let ψ be a coloring of G' with $t-1$ colors (namely, colors $1, \ldots, t-1$) and maximum number of colors with dominant vertices. By Lemma 3.1.3, all the dominant vertices are in C^*. We can extend ψ to a t-coloring of G by defining $\psi(c_1'') = t$. All dominant vertices in $\{c_2, \ldots, c_k\}$ are still dominant. If there were less than k dominant vertices, we may assume by symmetry of G' that they were in $\{c_2, \ldots, c_k\}$. If there were k dominant vertices in G', vertex c_1 is no longer dominant, still $c_2, \ldots, c_k$ are dominant, and we know that, if $\varphi(c_1') \neq \varphi(c_1'')$, then in G there cannot be more than $k-1$ colors with dominant vertices. So, in that case, $dom_G[k+s] = \min\{k-1, dom_{G'}[k+s-1]\}$. Since G' is type 2, $dom_G[k+s] = \min\{k-1, 2k-2s+2\}$.

So, if $2s \leq k$, then $dom_G[k+s] = k$ and the optimum is attained by a coloring where c_1' and c_1'' receive the same color. If $2s > k$, then $dom_G[k+s] = \min\{k-1, 2k-2s+2\}$ and the optimum is attained by a coloring where c_1' and c_1'' receive different colors. $\diamondsuit$

Claim 6. If G is type 8, then $dom_G[k+1] = k+1$; then $dom_G[k+s] = k$ for $2 \leq s \leq \lfloor \frac{k+1}{2} \rfloor$; $dom_G[k+s] = k-1$ for $s = \frac{k+2}{2}$ (when k is even); $dom_G[k+s] = 2k-2s+2$ for $\lfloor \frac{k+3}{2} \rfloor \leq s \leq k$; and $dom_G[t] = 0$ for $t > 2k$.

Since $\chi(G) = k+1$, then $dom_G[k+1] = k+1$. Since the maximum degree of G is $2k-1$, it is clear that $dom_G[t] = 0$ for $t > 2k$. Let $t = k+s$ with $2 \leq s \leq k$ and let φ be a t-coloring of G with maximum number of colors with dominant vertices. For $i \geq 2$, vertex c_i will be dominant if and only if color $\varphi(s_i)$ is used by some other vertex in G, and vertex c_1 will be dominant if and only if colors $\varphi(s_1')$ and $\varphi(s_1'')$ are used by some other vertices in $(c_1, s_2, \ldots, s_k)$. We may assume without loss of generality that $\varphi(c_i) = i$, for $i = 1, \ldots, k$, and that vertices in S^* use colors $k+1, \ldots, k+s$. If some vertex s_i uses a color at most k, we can always recolor it with a color from $k+1, \ldots, k+s$ that is already used in S^*. Since $s \geq 2$, we can do it also for s_1' and s_1''. If $2s \leq k+1$, we can assign colors $k+1, \ldots, k+s$ to vertices in S^*, repeating each of them at least once, in such a way that all the vertices in C^* are dominant. If $2s > k+1$, this is not possible. Since $\varphi(s_1') \neq \varphi(s_1'')$ and all the colors $k+1, \ldots, k+s$ are used in S^*, we may assume without loss of generality that $\varphi(s_1') = k+1$, $\varphi(s_1'') = k+2$, and $\varphi(s_i) = k+1+i$ for $i = 2, \ldots, s-1$ (when $s \geq 3$). To each of the $k+1-s$ remaining vertices we can assign different colors from $k+1, \ldots, k+s$. If we assign color $k+1+i$ to vertex s_j, with $s \leq j \leq k$ and $2 \leq i \leq s-1$, both c_i and c_j become dominant. If we assign color $k+1$ (resp. $k+2$) to some vertex s_j with $s \leq j \leq k$, then c_j will be dominant but c_1 will be dominant only if some other vertex $s_{j'}$, $s \leq j' \leq k$, receives $k+2$ (resp. $k+1$). So, as we have less than s remaining vertices, the optimum $2(k+1-s)$ is attained by assigning to $s_s, \ldots, s_k$ different colors from $k+3$ to $k+s$ when $k+1-s \leq s-2$. The last case is when $k+1-s = s-1$, that is, k is even and

$2s = k+2$. In this case we can assign to $s_s, \ldots, s_{k-1}$ different colors from $k+3$ to $k+s$ and to vertex s_k color $k+1$. In this case, all the vertices of C^* but c_1 are dominant, and this is optimal. ♢

Claim 7. If G is type 10, then $dom_G[k+s] = k$ for $0 \leq s \leq \lfloor \frac{k+1}{2} \rfloor$; $dom_G[k+s] = k-1$ for $s = \frac{k+2}{2}$ (when k is even); $dom_G[k+s] = 2k - 2s + 2$ for $\lfloor \frac{k+3}{2} \rfloor \leq s \leq k$; and $dom_G[t] = 0$ for $t > 2k$.

Since $\chi(G) = k$, then $dom_G[k] = k$. A coloring with $k+1$ colors and k dominant vertices is obtained by giving colors $1, \ldots, k$ to vertices in C^* and color $k+1$ to each vertex in S^*. Since the maximum degree of G is $2k-1$, it is clear that $dom_G[t] = 0$ for $t > 2k$. The arguments for $k+2 \leq s \leq 2k$ are very similar to those in the proof of Claim 6, and are omitted. ♢

In all the cases, given the type of the graph, the dominance sequence can be computed in linear time. The type of the graph can be also determined in linear time [38]. □

Theorem 3.3.2. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. Let $G = G_1 \cup G_2$ and $t \geq \chi(G)$. Then*

$$\mathrm{dom}_G[t] = min\{t, \mathrm{dom}_{G_1}[t] + \mathrm{dom}_{G_2}[t]\}$$

Theorem 3.3.3. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. Let $G = G_1 \vee G_2$ and $\chi(G) \leq t \leq |V(G)|$. Let $a = max\{\chi(G_1), t - |V(G_2)|\}$ and $b = min\{|V(G_1)|, t - \chi(G_2)\}$. Then $a \leq b$ and*

$$\mathrm{dom}_G[t] = max_{a \leq j \leq b}\{\mathrm{dom}_{G_1}[j] + \mathrm{dom}_{G_2}[t-j]\}$$

Theorem 3.3.4. *Let $G_1 = (V_1, E_1)$ be a quasi-urchin or a quasi-starfish of size k and $G_2 = (V_2, E_2)$ be a graph such that $V_1 \cap V_2 = \emptyset$, $V_2 \neq \emptyset$. Let $G = G_1 \veebar G_2$. Then, the following statements hold.*

i. If G_1 is type 1,2,7,9 or 10, then
  - *a.* $\mathrm{dom}_G[k+r] = k + \mathrm{dom}_{G_2}[r]$, *for* $\chi(G_2) \leq r \leq |V_2|$;
  - *b.* $\mathrm{dom}_G[k + |V_2| + s] = \mathrm{dom}_{G_1}[k+s]$, *for* $1 \leq s \leq |V_1| - k$.

ii. If G_1 is type 3 or 4, then
  - *a.* $\mathrm{dom}_G[k+1+r] = k + 1 + \mathrm{dom}_{G_2}[r]$, *for* $\chi(G_2) \leq r \leq |V_2|$;
  - *b.* $\mathrm{dom}_G[k+1+|V_2|+s] = \mathrm{dom}_{G_1}[k+1+s]$, *for* $1 \leq s \leq |V_1| - k - 1$.

iii. If G_1 is type 5 or 6, then
  - *a.* $\mathrm{dom}_G[k + \chi(G_2)] = k + \chi(G_2)$;
  - *b.* $\mathrm{dom}_G[k+r] = \max\{k + \mathrm{dom}_{G_2}[r], k - 1 + \mathrm{dom}_{G_2}[r-1]\}$, *for* $\chi(G_2) < r \leq |V_2|$;
  - *c.* $\mathrm{dom}_G[k+1+|V_2|] = \max\{k, k-1+\mathrm{dom}_{G_2}[|V_2|]\}$;
  - *d.* $\mathrm{dom}_G[k+|V_2|+s] = \mathrm{dom}_{G_1}[k+s]$, *for* $2 \leq s \leq |V_1| - k$.

iv. If G_1 is type 8, then
  - *a.* $\mathrm{dom}_G[k+r] = k + \mathrm{dom}_{G_2}[r]$, *for* $\chi(G_2) \leq r \leq |V_2|$;
  - *b.* $\mathrm{dom}_G[k+1+|V_2|] = k$;
  - *c.* $\mathrm{dom}_G[k+|V_2|+s] = \mathrm{dom}_{G_1}[k+s]$, *for* $2 \leq s \leq |V_1| - k$.

Proof. Recall that $dom[\chi(G)] = \chi(G)$, and that the chromatic number of each type of quasi-starfish or quasi-urchin is described in Lemma 3.1.8. Let (C^*, S^*) be the p-

partition of G_1. Notice first that in any coloring of G, the set of colors used by V_2 and C^* are disjoint. Let φ be a coloring of G with t colors, $t > \chi(G)$. Vertices in S^* are either simplicial or have degree at most $\chi(G) - 1$ (recall that $V_2 \neq \emptyset$). So no vertex in S^* can be dominant. If some vertex of S^* has a color that is used neither in V_2 nor in C^*, then no vertex in V_2 is dominant. We start the case analysis. If G_1 is type 1,2,7,9 or 10, then C^* is a clique of size k. Every vertex in C^* is dominant when the colors used by S^* are used also in $C^* \cup V_2$, and they are still dominant if we consider φ restricted to $G[V_2 \cup C^*]$. By Theorem 3.3.3, $dom_G[k+r] = k + dom_{G_2}[r]$, for $\chi(G_2) \leq r \leq |V_2|$. If $t > k + |V_2|$, at least some color must be used only in S^*. So the only candidates to be dominant vertices are vertices in C^*. Since they are adjacent to all the vertices in V_2, we may assume that no vertex in S^* uses a color used in V_2, and each vertex of C^* is dominant if and only if it is dominant in $G[V_1]$, so $dom_G[k + |V_2| + s] = dom_{G_1}[k + s]$, for $1 \leq s \leq |V_1| - k$ $(*)$. If G_1 is type 3 or 4, the analysis is the same but taking into account that C^* is a clique of size $k + 1$. If G_1 is type 5 or 6, then C^* is not a clique. We may assume that the original set was $C = \{c_1, \ldots, c_k\}$ and vertex c_1 was replaced by two false twins c'_1, c''_1. Item *iii.a* holds because $\chi(G) = k + \chi(G_2)$. Most of the observations for the previous cases still hold. So, when $\chi(G_2) < t - k \leq |V_2| + 1$, we have two possibilities to color C^*: we can either use k colors, being $\varphi(c'_1) = \varphi(c''_1)$, and in that case k vertices of different colors will be dominant in C^*, or use $k + 1$ colors and, by Lemma 3.1.1, only $k - 1$ vertices in C^* will be dominant. This leads to the expressions *iii.b* and *iii.c*. Finally, when $t > k + |V_2| + 1$, at least some color must be used only in S^*. The analysis in $(*)$ leads to the expression *iii.d*. Finally, if G_1 is type 8, then C^* is a clique of size k but $\chi(G_1) = k + 1$. In this case, if $\chi(G_2) \leq r \leq |V_2|$, necessarily one color in V_2 will be used also in S^*, but the analysis is the same as in case *i.a*. Also the case *iv.c* is similar to *i.b*. The only difference is when $t = k+1+|V_2|$. We cannot say that $dom_G[k + 1 + |V_2|] = dom_{G_1}[k + 1] = k + 1$, because we know that we have dominant vertices only in C^*, so $dom_G[k + 1 + |V_2|] \leq k$. A coloring with k dominant vertices in C^* is attainable by giving colors $1, \ldots, k$ to vertices in C^*, color $k + 1$ to vertices in $S^* \setminus \{s''_1\}$, color $k + 2$ to s''_1, and colors $k + 2, \ldots, k + 1 + |V_2|$ to vertices in V_2. □

Theorem 3.3.5. *The dominance vector and the b-chromatic number of a P_4-tidy graph G can be computed in $O(n^3)$ time.*

Proof. The previous results give a dynamic programming algorithm to compute the dominance sequence of a P_4-tidy graph from its decomposition tree, that can be computed in linear time [38]. By Theorem 3.3.2, Theorem 3.3.3, Theorem 3.3.4, Theorem 3.1.7, Proposition 3.1.1 and the fact that P_4-tidy graphs are hereditary, we can compute recursively the dominance vector and consequently the b-chromatic number of G in $O(n^3)$ time. Indeed, if $G = G_1 \cup G_2$, by Theorem 3.3.2, the value for $dom_G[t]$ is obtained from $dom_{G_1}[t]$ and $dom_{G_2}[t]$ directly. By Theorem 3.3.4, the same case holds for $G = G_1 \veebar G_2$. If $G = G_1 \vee G_2$, we must examine at most n values of j for each value of t, by Theorem 3.3.3. We have at most n of these reduction steps, because in each case we must compute two disjoint subgraphs. The base case, computing the dominance sequence of the trivial graph and the five elementary subgraphs in the decomposition, can be done in $O(1)$ by Lemma 3.3.1. So the total computation time is $O(n^3)$. Once we have computed the dominance sequence of G, we obtain the b-chromatic number as the maximum value t such that $dom_G[t] = t$. □

3.4 b-monotonicity in P_4-tidy graphs

In this section, we will show that P_4-tidy graphs are b-monotonic. To this end, we will prove the following property.

Theorem 3.4.1. *For every P_4-tidy graph G, every induced subgraph H of G and every $t \geq \chi(G)$, $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ holds.*

We first state some necessary results.

Lemma 3.4.2. *Let G be a P_5, a $\overline{P_5}$, a C_5, a quasi-urchin or a quasi-starfish. Then, for every $t \geq \chi(G)$ and every vertex v of G, $\mathrm{dom}_{G-\{v\}}[t] \leq \mathrm{dom}_G[t]$ holds.*

Proof. The cases P_5, $\overline{P_5}$ and C_5 are easy to verify. Let $G = (C^*, S^*)$ be a quasi-urchin or quasi-starfish of size k. Let (C, S) be the p-partition of the urchin or starfish, $S = \{s_1, \dots, s_k\}$, $C = \{c_1, \dots, c_k\}$. If a vertex in S (resp. C) was replaced by two vertices, we will assume that the vertex was s_1 (resp. c_1) and that it was replaced by vertices s'_1, s''_1 (resp. c'_1, c''_1). Let $t \geq \chi(G)$, and let v be a vertex of G. Let φ be a t-coloring of $G - \{v\}$ that maximizes the number of color classes with dominant vertices. Suppose first that v is a leg of G and either G is not type 8 or v is different from s'_1, s''_1. Then φ can be extended to a t-coloring of G with the same number of dominant vertices by giving to v the color of some vertex in the body non-adjacent to it. If G is type 8 and $v = s'_1$, since $t \geq \chi(G) = k+1$, we can give to s'_1 either a color that is not used in the body of G or the color $\varphi(c_1)$ (depending on whether $\varphi(s''_1) = \varphi(c_1)$ or not). Now, suppose that v is a vertex in the body of G. If v has a false twin, we can color v with the color used by its false twin. Otherwise, since $t \geq \chi(G)$, there is some color c that is not used in the body of G. We will extend φ to a t-coloring of G with at least the same number of dominant vertices by setting $\varphi(v) = c$. If some leg w of G adjacent to v was colored c, then all its neighbors are also neighbors of v, so we can recolor w with the color of some vertex in the body non-adjacent to it, and all dominant vertices will still be dominant. The only case in which we cannot do this is when G is type 8, v is not c_1, one of s'_1, s''_1 uses color c and the other one uses color $\varphi(c_1)$. But, in that case, since $t \geq \chi(G) = k + 1$, there are in fact at least two colors c, c' not used in the body of G. So we can give color c' to v, and recolor as mentioned above all the legs adjacent to it (note that neither s'_1 nor s''_1 use c' in the case we are dealing with). Hence, $dom_{G-\{v\}}[t] \leq dom_G[t]$. □

Lemma 3.4.3. *Let $G_1 = (V_1, E_1)$ be a quasi-starfish or a quasi-urchin and $G_2 = (V_2, E_2)$ be a b-continuous graph such that $V_1 \cap V_2 = \emptyset$ and, for every $t \geq \chi(G_2)$ and every induced subgraph H of G_2, $\mathrm{dom}_H[t] \leq \mathrm{dom}_{G_2}[t]$. Let $G = G_1 \veebar G_2$. Then, for every $t \geq \chi(G)$ and every vertex v of G, $\mathrm{dom}_{G-\{v\}}[t] \leq \mathrm{dom}_G[t]$ holds.*

Proof. If $t = \chi(G)$ the statement is clearly true. Let $t > \chi(G)$, and let v be a vertex of G. Let φ be a t-coloring of $G - \{v\}$ that maximizes the number of color classes with dominant vertices. We will extend φ to a t-coloring of G with the same number of color classes with dominant vertices. Let (C^*, S^*) be the p-partition of G_1. Notice that, since $t > \chi(G) \geq \chi(G - \{v\})$, no vertex in S^* is dominant.

Suppose first that v is a vertex of S^*. We can extend φ by giving to v a color not used by any of its neighbors (it is always possible because $t > \chi(G)$).

Suppose now that v is a vertex of V_2. If $|V_2| = 1$ then the lemma holds by Theorem 3.3.4 and the claims in the proof of Lemma 3.3.1. If $|V_2| > 1$, let r be the number of colors used by $V_2 - \{v\}$ in φ. If $r \geq \chi(G_2)$, since $dom_{G_2}[r] \geq dom_{G_2 - \{v\}}[r]$, we can replace φ restricted to $V_2 - \{v\}$ by an r-coloring of G_2 with $dom_{G_2}[r]$ color classes with dominant vertices, thus obtaining a t-coloring of G with at least the same dominant color classes as before. Otherwise, $r = \chi(G_2 - \{v\}) = \chi(G_2) - 1$. Since $t > \chi(G)$, it follows that $t - r \geq \chi(G_1)$. Notice that the equality can hold only if G_1 is type 8. Then we can replace φ restricted to $V_2 - \{v\}$ by an $(r + 1)$-coloring of G_2 and, by Lemma 3.3.1, φ restricted to V_1 by a coloring of G_1 with at most $t - r - 1$ new colors and at least the same dominant color classes as before (if G_1 is type 8 and $t - r = \chi(G_1)$, we can assign to one of the true twin vertices in S^* a color used in V_2).

Finally, suppose that v is a vertex in C^*. If v has a false twin v' in $C^* - \{v\}$, we are done by setting $\varphi(v) = \varphi(v')$. If there are two false twins w, w' in $C^* - \{v\}$ using different colors, we can assign to v color $\varphi(w')$ and to w' color $\varphi(w)$ (possibly recoloring in a suitable way vertices in S^*), obtaining a t-coloring of G with at least the same dominant color classes as before. Otherwise, v is adjacent to all vertices in $C^* - \{v\}$ and they are colored with $\chi(G[C^* - \{v\}])$ colors. Let r be the number of colors used by V_2 in φ. If φ restricted to V_2 is not a b-coloring, we can eliminate one color class from V_2 without decreasing the number of color classes with dominant vertices, and give that color to v, that will be adjacent to all the vertices that were dominant, thus obtaining the desired t-coloring for G. If φ restricted to V_2 is a b-coloring and $r > \chi(G_2)$, since G_2 is b-continuous, we can replace φ restricted to V_2 by a b-coloring of G_2 with $r - 1$ colors, thus giving the remaining color to v as before. Finally, by Lemma 3.1.9 and being $t > \chi(G)$, if $r = \chi(G_2)$ then $t - r \geq \chi(G_1)$. In that case, it is easy to see that we can replace φ restricted to V_1 by a $t - r$-coloring of G_1, maintaining or increasing the number of color classes with dominant vertices, thus obtaining the desired t-coloring of G. □

Lemma 3.4.4. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$, and let $G = G_1 \cup G_2$. Assume that for every $t \geq \chi(G_i)$ and every induced subgraph H of G_i we have $\mathrm{dom}_H[t] \leq \mathrm{dom}_{G_i}[t]$, for $i = 1, 2$. Then, for every $t \geq \chi(G)$ and every induced subgraph H of G, $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ holds.*

Lemma 3.4.5. [16] *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two b-continuous graphs such that $V_1 \cap V_2 = \emptyset$, and let $G = G_1 \vee G_2$. Assume that for every $t \geq \chi(G_i)$ and every induced subgraph H of G_i we have $\mathrm{dom}_H[t] \leq \mathrm{dom}_{G_i}[t]$, for $i = 1, 2$. Then, for every $t \geq \chi(G)$ and every induced subgraph H of G, $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ holds.*

Lemma 3.4.6. [16] *Let G be a graph. The maximum value of* $\mathrm{dom}_G[t]$ *is attained in* $t = \chi_b(G)$.

Proof of Theorem 3.4.1. Let us consider a minimal counterexample for the theorem, that is, a P_4-tidy graph G and an induced subgraph H of G such that $dom_H[t] > dom_G[t]$ for some $t \geq \chi(G)$, but such that $dom_{H_2}[t] \leq dom_{H_1}[t]$ for every induced subgraph H_1 of H, every induced subgraph H_2 of H_1 and every $t \geq \chi(H_1)$. By Lemmas 3.4.4 and 3.4.5, G is neither the union nor the join of two smaller graphs. Let $W = V(G) \setminus V(H)$, namely, $W = \{w_1, \ldots, w_s\}$. If we consider the sequence of graphs defined by $G_0 = G$, $G_i = G_{i-1} - \{w_i\}$ for $1 \leq i \leq s$, it turns out that $G_s = H$. Since $dom_H[t] > dom_G[t]$, for some $i \geq 1$, it holds $dom_{G_i}[t] > dom_{G_{i-1}}[t]$. Since the counterexample was minimal, it should be $i = 1$, thus $H = G - \{w\}$ for some vertex w of G. By Lemma 3.4.2, Theorem 3.2.5 and Lemma 3.4.3, Theorem 3.1.7 and Proposition 3.1.1, such a counterexample does not exist. □

Corollary 3.4.6.1. *P_4-tidy graphs are b-monotonic.*

Proof. Since P_4-tidy graphs are hereditary, it suffices to show that given a P_4-tidy graph G, $\chi_b(G) \geq \chi_b(H)$ for every induced subgraph H of G. Let G be a P_4-tidy graph, and let H be and induced subgraph of G. If $\chi_b(H) < \chi(G)$, then $\chi_b(H) < \chi_b(G)$. Otherwise, by Theorem 3.4.1, $\chi_b(H) = dom_H[\chi_b(H)] \leq dom_G[\chi_b(H)]$ and, by Lemma 3.4.6, $dom_G[\chi_b(H)] \leq dom_G[\chi_b(G)] = \chi_b(G)$ implying that $\chi_b(G) \geq \chi_b(H)$. □

3.5 Resumen del capítulo

Un *b-coloreo* de un grafo es un coloreo tal que cada clase color admite un vértice adyacente a por lo menos un vértice de cada una de las demás clases color. El *número b-cromático* de un grafo G, denotado por $\chi_b(G)$, es el máximo número t tal que G admite un b-coloreo con t colores. El problema de b-coloreo fue introducido por R. W. Irving y D. F. Manlove [56] al considerar coloreos válidos que fueran minimales con respecto a un orden parcial definido sobre el conjunto de todas las particiones del conjunto de vértices de G. Los autores demostraron en ese trabajo que determinar el número b-cromático de un grafo G es NP-hard, pero puede resolverse en tiempo polinomial para árboles. Kratochvíl, Tuza y Voigt mostraron en [77] que determinar $\chi_b(G)$ es NP-hard aún si G es un grafo bipartito conexo, pero puede ser resuelto en tiempo polinomial para algunas familias de grafos bipartitos. Bonomo et al. mostraron que el problema es NP-hard también para grafos co-bipartitos, pero polinomial para tree-cographs [17]. Corteel et al. [30] mostraron que el problema es también difícil de aproximar en tiempo polinomial con un factor de $\frac{120}{113} - \epsilon$, para cualquier $\epsilon > 0$, salvo que $P = NP$.

El comportamiento del número b-cromático puede ser sorprendente. A diferencia del coloreo clásico, los valores de k para los cuales el grafo admite un b-coloreo con k colores no necesariamente forman un intervalo en el conjunto de los enteros. De hecho, cualquier subconjunto finito de $\mathbb{N}_{\geq 2}$ constituye el conjunto de valores para los cuales existe un b-coloreo en algún grafo [9]. Un grafo G es *b-continuo* si admite un b-coloreo con t colores, para cada $t = \chi(G), \ldots, \chi_b(G)$. En [68] se demuestra que los grafos cordales y algunos grafos planares son b-continuos.

Otra propiedad atípica es que el número b-cromático puede incrementarse al tomar subgrafos inducidos. Un grafo G es *b-monótono* si $\chi_b(H_1) \geq \chi_b(H_2)$ para cada subgrafo inducido H_1 de G, y cada subgrafo inducido H_2 de H_1 [16].

Un *cografo* es un grafo que no contiene a P_4 como subgrafo inducido. Esta clase de grafos fue descubierta independientemente por varios autores. Corneil et al. propusieron un algoritmo lineal de reconocimiento basado en una descomposición única del cografo [27]. Se definieron varias generalizaciones de cografos en la literatura, como los grafos P_4-sparse [51], P_4-lite [60], P_4-extensibles [62] y los grafos P_4-reducibles [61]. Una clase de grafos que generaliza a todos ellos es la clase de los grafos P_4-tidy [38]. Sea G un grafo y A un P_4 en G. Un *partner* de A es un vértice v en $G - A$ tal que $A \cup \{v\}$ induce por lo menos dos P_4s en G. Un grafo G es *P_4-sparse* si ningún P_4 inducido tiene un partner y es *P_4-tidy* si cada P_4 inducido tiene a lo sumo un partner. Otra generalización de los grafos P_4-sparse son los grafos$(q, q-4)$. Un grafo es $(q, q-4)$ si ningún conjunto de a lo sumo q vértices induce más de $q - 4$ P_4's distintos [4]. En [24] Campos, Linhares Sales, Maia y Sampaio obtuvieron un algoritmo polinomial para computar el número b-cromático de los grafos $(q, q - 4)$, para q fijo. No hay relación de inclusión entre las clases P_4-tidy y $(q, q - 4)$.

A continuación enunciamos los principales resultados de este capítulo.

Teorema: El número b-cromático de un grafo P_4-tidy puede ser computado en $O(n^3)$ time.

Con este teorema extendemos los resultados presentados en [16] para la clase de grafos P_4-sparse.

Teorema: Los grafos P_4-tidy son b-continuos.

Teorema: Los grafos P_4-tidy son b-monótonos.

Los resultados en este capítulo fueron publicados en [11].

CHAPTER 4

The b-chromatic index of the direct product of graphs

We continue in this chapter the study of the edge version of the b-vertex coloring and the b-chromatic number introduced by Jakovac and Peterin in [58], namely the b-edge coloring and the b-chromatic index, respectively. A *b-edge coloring* of a graph G is a proper edge coloring of G such that each color class contains an edge that has at least one incident edge in every other color class. The *b-chromatic index of a graph G* is the largest integer $\chi'_b(G)$ for which G has a b-edge coloring with $\chi'_b(G)$ colors. We say that that a b-edge coloring with $\chi'_b(G)$ colors *realizes* $\chi'_b(G)$.

Intuitively, for a b-edge coloring to be possible on a graph G, we need to have enough edges of high enough degree, at least one in each color class. Let $e_1, \ldots, e_n$ be a sequence of edges, such that $d(e_1) \geq \cdots \geq d(e_n)$ where $d(e_i)$ denotes the degree of e_i. Then $m'(G) = max\{i : d(e_i) \geq i-1\}$ is an upper bound for $\chi'_b(G)$. In [58], the authors determined the b-chromatic index of trees, and gave conditions for graphs that have b-chromatic index strictly less than $m'(G)$, as well as conditions for which $\chi'_b(G) = m'(G)$. They proved further that $\chi'_b(G) = 5$ for connected cubic graphs, with only four exceptions: K_4, $K_{3,3}$, the prism over K_3, and the cube Q_3. The problem of computing the b-chromatic index was shown to be NP-complete by Lima et al. in [78].

The *direct product* $G \times H$ of graphs G and H has vertex set $V(G) \times V(H)$; two vertices (g,h) and (g',h') are adjacent in $G \times H$ if they are adjacent in both coordinates, i.e. $gg' \in E(G)$ and $hh' \in E(H)$. See Figure 4.1 for an example. If $e = (g,h)(g',h') \in E(G \times H)$, let $p_G(e) = gg'$ and $p_H(e) = hh'$ be the *projection* of edge e over G and H, respectively. Unfortunately, it appears in the literature also as tensor product, Kronecker product, cross product, and categorical product, among other denominations. The direct product seems to be the most elusive among all four standard products (Cartesian, strong, direct and lexicographic). The reason for this is

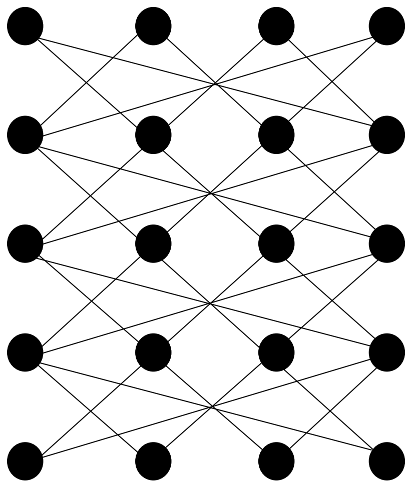


Figure 4.1: Illustration of the graph $P_5 \times C_4$.

the fact that each edge of $G \times H$ projects to an edge in both factors, which is not the case on other products. Even basic graph properties, such as connectedness are non trivial for the direct product. Indeed, $G \times H$ need not be connected, even if both factors are. It can be shown that if both factors are bipartite and connected, then there are exactly two components in the direct product (see [98]). Since its initial formulation by Weichsel [98] in 1962, the direct product of graphs has been extensively studied in the areas of graph coloring, graph recognition and decomposition, graph embeddings, matching theory and stability in graphs (see the book [42] for a comprehensive survey). There is a problem, though, that is regarded as the main open question for the direct product, and is related to graph coloring. In 1966, Hedetniemi [49] conjectured that for all graphs G and H, $\chi(G \times H) = min\{\chi(G), \chi(H)\}$. This famous conjecture has resisted the efforts of researchers until the present day.

In this chapter, we describe bounds for the b-chromatic index of the direct product of a graph G and a regular graph H admitting a partition of its edges into perfect matchings. Further, we present exact results of χ'_b for the direct product of some connected regular graphs. Determining the b-chromatic index of a graph can be very tedious already for small examples. For this reason we also develop a simple integer linear programming model for the problem. With this method and all previous results we were able to produce exact values of χ'_b for the direct product of paths and for the direct product of cycles. The results in this work were proposed in [74].

4.1 Bounds for $\chi'_b(G \times H)$

A *one-factor* or a *perfect matching* of a graph G is a set of independent edges of G that meet every vertex of G. Clearly, a graph with a one-factor has an even number of vertices. A *one-factorization* of G is a partition of $E(G)$ into one-factors. Thus, in a one-factorization of G, every edge belongs to exactly one one-factor. Evidently

G must be regular with an even number of vertices if it has a one-factorization. The basic examples of graphs with one factorization are even cycles and hypercubes. Graph products form a rich field for one factorizations, see for instance Section 30.1 of [42].

The following result shows a somewhat surprising connection between graphs with one-factorizations and the b-chromatic index of direct products.

Theorem 4.1.1. *Let G be a graph and H an r-regular graph. If H has a one-factorization, then*

$$\chi'_b(G \times H) \geq r\chi'_b(G).$$

Proof. Let G be a graph with $\chi'_b(G) = k$ and let $c' : E(G) \to \{1, \ldots, k\}$ be a b-edge coloring that realizes $\chi'_b(G)$. If H is an r-regular graph with a one-factorization, then there are exactly r one factors that partition $E(H)$. Let $c'' : E(H) \to \{1, \ldots, r\}$ be a proper edge coloring where $c(e) = i$ whenever $e \in E(H)$ belongs to the ith one-factor of H. We will show that $c : E(G \times H) \to \{(1, 1), (1, 2), \ldots, (k, r)\}$ defined by

$$c(e) = (c'(p_G(e)), c''(p_H(e)))$$

is a b-edge coloring of $G \times H$.

First we show that c is a proper edge coloring. Suppose, on the contrary, that for two incident edges $e, f \in E(G \times H)$ we have $c(e) = c(f)$, where $e = (g, h)(g', h')$ and $f = (g, h)(g'', h'')$. Clearly $e \neq f$ implies that $g' \neq g''$ or $h' \neq h''$. Since $c(e) = (c'(gg'), c''(hh')) = (c'(gg''), c''(hh'')) = c(f)$, we have $c'(gg') = c'(gg'')$ and $c''(hh') = c''(hh'')$. If $g' \neq g''$, $c'(gg') = c'(gg'')$ is a contradiction with c' being a proper coloring of $E(G)$. Similarly, if $h' \neq h''$, $c''(hh') = c''(hh'')$ is a contradiction with c'' being a proper coloring of $E(H)$. Hence c is a proper coloring of $E(G \times H)$.

Next, let $e = (g, h)(g', h')$ be an edge of $G \times H$, where gg' is (without loss of generality) a 1-dominating edge in G for c'. We claim that e is a $c(e) = (1, c''(hh'))$ dominating edge in $G \times H$ for c. Let $g_1, \ldots, g_j$ be neighbors of g, such that gg_p, $p \in \{1, \ldots, j\}$, is assigned a color needed for e to be a 1-dominating edge. Similarly, let $g'_1, \ldots, g'_\ell$ be neighbors of g' such that $g'g'_t$, $t \in \{1, \ldots, \ell\}$, is also assigned a color needed for e to be 1-dominating. (Notice that $j + \ell = k - 1$.) Recall that coloring c'' is generated by a one-factorization of H, which yields for every vertex $h \in V(H)$ that all colors are present on edges incident with h. We may assume without loss of generality that $c''(hh') = 1$. Denote by $h_2, \ldots, h_r$ neighbors of h where $c''(hh_s) = s$, $s \in \{2, \ldots, r\}$, and by $h'_2, \ldots, h'_s$ neighbors of h' where $c''(h'h'_i) = s$, $s \in \{2, \ldots, r\}$. Edges $(g, h)(g_p, h_s)$ and $(g', h')(g'_t, h'_s)$, $p \in \{1, \ldots, j\}$, $t \in \{1, \ldots, \ell\}$ and $s \in \{2, \ldots, r\}$, are all incident with e and they have all colors of the set $\{2, \ldots, k\} \times \{2, \ldots, r\}$. Similarly, edges $(g, h)(g_p, h')$, $p \in \{1, \ldots, j\}$, and $(g', h')(g'_t, h)$, $t \in \{1, \ldots, \ell\}$, have all colors of the set $\{2, \ldots, k\} \times \{1\}$. Finally, edges $(g, h)(g', h_s)$, $s \in \{2, \ldots, r\}$, are assigned all colors from $\{1\} \times \{2, \ldots, r\}$. (Notice that edges $(g', h')(g, h'_s)$, $s \in \{2, \ldots, r\}$, are colored with the same colors.) Hence e with $c(e) = (1, 1)$ has all colors from $\{1, \ldots, k\} \times \{1, \ldots, r\} - \{(1, 1)\}$ on edges incident to e, which yields that e is a $(1, 1)$-dominating edge, so the proof is completed. □

The lower bound of Theorem 4.1.1 seems to present its best when r is small. The reason for this can be found in the last part of the proof where edges $(g', h')(g, h'_s)$, $s \in \{2, \ldots, r\}$, are colored with the same colors. Hence here we have $r-1$ duplicated colors and this may lead to strict inequality.

Corollary 4.1.1.1. *Let $n \geq 2$ be an integer. If G is an r-edge regular graph with $\chi'_b(G) = r+1$, then*

$$2r + 2 \leq \chi'_b(G \times C_{2n}) \leq 2r + 3.$$

Proof. By Theorem 4.1.1 we get the lower bound, while the upper bound is the trivial upper bound $m'(G \times C_{2n})$. □

While the reader might find that this corollary holds only for a small number of graphs, let us recall that there exists only a finite number of r-regular graphs with $\chi_b(G) < m(G)$ (see [20]). If G is an r-edge regular graph, then $\mathcal{L}(G)$ is an r-regular graph. "Usually" this means that $\chi'_b(G) = r + 1$; in particular, it holds for all r-edge regular graphs with at least $2r^3$ edges (Theorem 2.2 in [20]). But even if the number of edges is smaller than $2r^3$, one can expect that most problems will occur when the number of edges is small. The computational results from [37] for the b-chromatic number of small regular graphs indicate such a conclusion.

On the other hand, if one wishes to generalize Theorem 4.1.1 to all direct products, one can observe from the proof that the one-factorization of a graph H was needed "only" to obtain an edge coloring of H with the following property: there must exist an edge $e = uv$, called *symmetric*, in every color class for which every endvertex has all colors on edges incident to it. We call a proper coloring with this property an *edge symmetric coloring* of H. Unfortunately, edge symmetric colorings do not exist for every graph H. For this observe odd cycles. Since every edge symmetric coloring is a proper coloring, we need at least $\chi'(C_{2k+1}) = 3$ colors. However this is not possible, since C_{2k+1} is 2-regular. More generally, an edge symmetric graph coloring of H can contain at most $\Delta(H)$ colors, since a symmetric edge e can have at most $\Delta(H) - 1$ edges with different colors incident to each endvertex plus the color of e. Thus no class 2 graph has an edge symmetric coloring. Many class 1 graphs may also lack an edge symmetric coloring, too, since we need for such a coloring at least $\Delta(H)$ edges in which all endvertices have degree $\Delta(H)$.

Nevertheless, with the concept of edge symmetric colorings we can extend the bound from Theorem 4.1.1 to many other direct products, in particular to products in which both factors are non-regular. One example of graphs with an edge symmetric coloring are paths P_n, $n \geq 5$. Another small example is a path P_6 with edges colored 3-2-1-3-2 in turn and in each inner vertex an additional edge attached, colored 1-3-2-1 in turn.

Theorem 4.1.2. *Let G and H be graphs. If H contains an edge symmetric coloring, then*

$$\chi'_b(G \times H) \geq \chi'_b(G)\Delta(H).$$

Proof. The proof is the same as the proof of Theorem 4.1.1 in which we replace coloring c'' generated by a one-factorization of H by an edge symmetric b-coloring $\overline{c}$ of H. □

4.2 On the b-chromatic index of direct products of some regular graphs

First we recall a result from [58] for graphs with $diam(G) \geq 4$. For a vertex v of G, let $S_2(v)$ be a set of all vertices of G that are at distance 2 from v. We say that a graph G is of *class 1* if $\chi'(G) = \Delta(G)$ and of *class 2* if $\chi'(G) = \Delta(G) + 1$. We define the graph $G[v]$ as the subgraph of G induced by $N(v) \cup S_2(v)$.

Theorem 4.2.1. *[58] Let G be an r-regular graph with $diam(G) \geq 4$ and let u and v be two vertices at distance at least 4. If $G[u]$ and $G[v]$ are class 1 graphs with $\Delta(G[u]) = \Delta(G[v]) = \Delta(G) - 1$, then*

$$\chi'_b(G) = 2r - 1.$$

The condition $\Delta(G[v]) = r - 1$ implies that each vertex from $S_2(v)$ has a neighbor at distance 3 from v. If this condition is not met in an r-regular bipartite graph G, then there exists $u \in S_2(v)$ for which $N(u) = N(v)$. However this is not necessarily the case for nonbipartite graphs. Notice also that for every bipartite graph G and any vertex v of G, the graph $G[v]$ is bipartite (bipartition is induced with $N(v)$ and $S_2(v)$) with $\Delta(G[v]) = r-1$. Recall that every bipartite graph is class 1 graph by König's Theorem. We will now deduce the conditions of Theorem 4.2.1 for the direct product with the help of the following lemmas. We recall first the distance formula for the direct product (see [72]),

$$d_{G\times H}((g,h),(g',h')) = \min\{\max\{d^e_G(g,g'), d^e_H(h,h')\}, \max\{d^o_G(g,g'), d^o_H(h,h')\}\}.$$

Here $d^e_G(g,g')$ means the length of a shortest walk of even length between g and g' in G and $d^o_G(g,g')$ the length of a shortest odd walk between g and g' in G. If such a walk does not exist, we set $d^e_G(g,g')$ or $d^o_G(g,g')$ to be infinite.

Lemma 4.2.2. *Let G and H be connected graphs without triangles. If at least one of G and H is nonbipartite, then $diam(G \times H) \geq 4$.*

Proof. Since at least one factor is nonbipartite, $G \times H$ is connected. Let $g \in V(G)$ and $hh' \in E(H)$. Clearly $d^e_G(g,g) = 0$ and $d^o_H(h,h') = 1$. Since G has no triangle, the shortest odd cycle has length at least 5. Even if g lies on such a cycle it holds that $d^o_G(g,g) \geq 5$. Similarly, the shortest odd cycle of H has length at least 5. Even if both h and h' lie on such a cycle we have $d^e_H(h,h') \geq 4$. By the distance formula we get $4 \leq d_{G\times H}((g,h),(g,h')) < \infty$ and hence the result. □

If both graphs are bipartite we need a slightly different approach.

Lemma 4.2.3. *Let G and H be connected bipartite graphs. If $diam(G) \geq 4$ or $diam(H) \geq 4$, then both components of $G \times H$ have diameter at least 4.*

Proof. Let $g \in V(G)$ and $hh' \in E(H)$. Vertices (g, h) and (g, h') belong to different components of $G \times H$. Suppose without loss of generality that $diam(G) \geq 4$ and that $d_G(g, g') = 4$ for a vertex $g' \in V(G)$. Clearly $d^e_G(g, g') = 4$, $d^o_G(g, g') = \infty$ and $d^e_H(h, h) = 0$. By the distance formula we get $d_{G\times H}((g, h), (g', h)) = 4$. The proof for the component of (g, h') is analogous, and hence the result follows. □

The second condition of Theorem 4.2.1 is that for vertices u and v with $d_G(u, v) \geq 4$, graphs $G[u]$ and $G[v]$ must be class 1 graphs. The next lemma settles this problem. For this we need one of the well known facts about direct products, namely

$$N^{G\times H}(g, h) = N^G(g) \times N^H(h) \tag{4.1}$$

where $N^G(v)$ denotes the subgraph induced by the neighborhood of vertex v in a graph G. Moreover, it is also easy to see that

$$S_2^{G\times H}(g, h) \cup \{(g, h)\} = (S_2^G(g) \cup \{g\}) \times (S_2^H(h) \cup \{h\}) \tag{4.2}$$

holds for any graphs G and H without triangles.

Lemma 4.2.4. *Let G and H be connected graphs without triangles and let $(g, h) \in V(G \times H)$. If no five cycle contains g, then $G \times H[(g, h)]$ is a class 1 graph. In particular if G has no five cycles, then $G \times H[(g, h)]$ is a class 1 graph.*

Proof. If $G \times H$ is bipartite, then $G \times H[(g, h)]$ is bipartite as it is an induced subgraph of $G \times H$ and hence a class 1 graph by König's Theorem. Suppose that $G \times H$ is nonbipartite. Edges of an odd cycle of $G \times H$ project to edges in both factors, which yields an odd closed walk in both factors. Therefore both G and H are nonbipartite.

Let (g, h) be an arbitrary vertex of $G \times H$. The direct product of triangle free graphs is triangle free again. Hence $N^{G\times H}(g, h)$ induces an empty graph. Suppose that g is not on a five cycle in G. By expression (4.2) above, vertices of $S_2^{G\times H}(g, h)$ project to vertices in $S_2^G(g) \cup \{g\}$. But vertices of $S_2^G(g) \cup \{g\}$ induce a graph without edges, since there are no triangles in G and g is not on a five cycle. Therefore also $S_2^{G\times H}(g, h)$ induces an empty graph, $G\times H[(g, h)]$ is bipartite and thus a class 1 graph. (Notice that if h is not on a five cycle in H we can exchange the role of g and h by commutativity of the direct product and we are done again.) □

Notice that this lemma can also be true when both graphs contain five cycles. The smallest example is $C_5 \times C_5$, where $G[(g, h)]$ is isomorphic to a graph obtained from K_4 by a 1-subdivision of each edge on some fix four cycle and a 2-subdivision of the remaining two edges. Such a graph is easily colored by 3 colors and is hence a class 1 graph.

Finally we discuss the last condition of Theorem 4.2.1, which says that in an r-regular graph $\Delta(G[v]) = r - 1 = \Delta(G[u])$. With this additional lemma we close the discussion on the conditions of Theorem 4.2.1 for the direct product.

Lemma 4.2.5. *Let G and H be connected graphs, r_G- and r_H-regular, respectively, and let $(g,h) \in V(G \times H)$. If $r_G, r_H > 0$ and $\Delta(G[g]) = r_G - 1$ or $\Delta(H[h]) = r_H - 1$, then $\Delta(G \times H[(g,h)]) = r_G r_H - 1$.*

Proof. Clearly $G \times H$ is an $r_G r_H$-regular graph. Let g be a vertex of G that verifies $\Delta(G[g]) = r_G - 1$. Every vertex from $N^{G \times H}(g,h)$ has degree $r_G r_H - 1$ in $G \times H[(g,h)]$, since (g,h) is not in $G \times H[(g,h)]$ (for arbitrary $h \in V(H)$). Every vertex g' from $S_2^G(g)$ has a neighbor outside of $G[g]$ because $\Delta(G[g]) = r_G - 1$. Therefore every vertex (g', h') from $S_2^{G \times H}(g,h)$ has at least $r_H > 0$ neighbors outside of $G \times H[(g,h)]$, which yields $\Delta(G \times H[(g,h)]) = r_G r_H - 1$. By commutativity of the direct product we are done also when h is such a vertex in H, that $\Delta(H[h]) = r_H - 1$. □

Unfortunately the above lemmas differ in their assumptions, so that we need to be careful stating the following results.

Theorem 4.2.6. *Let G and H be connected graphs, r_G- and r_H-regular, respectively, and without triangles. Let additionally $gg' \in E(G)$, where g and g' do not lie on any five cycle and $\Delta(G[g]) = r_G - 1 = \Delta(G[g'])$. If at least one of G and H is nonbipartite, then*

$$\chi_b'(G \times H) = 2r_G r_H - 1.$$

Proof. Let $gg' \in E(G)$, such that g and g' do not lie on any five cycle. By Lemma 4.2.2 we have $diam(G \times H) \geq 4$. Moreover (g,h) and (g',h) are at distance at least 4 for any $h \in V(H)$. Lemma 4.2.4 implies that $G \times H[(g,h)]$ and $G \times H[(g',h)]$ are class 1 graphs and Lemma 4.2.5 that $\Delta(G \times H[(g,h)]) = r_G r_H - 1$ and $\Delta(G \times H[(g',h')]) = r_G r_H - 1$. By Theorem 4.2.1 the result follows. □

Theorem 4.2.7. *Let G and H be connected bipartite graphs, and r_G- and r_H-regular, respectively. Let $diam(G) \geq 4$. If there exists $g, g' \in V(G)$ with $d_G(g,g') \geq 4$ and $\Delta(G[g]) = r_G - 1 = \Delta(G[g'])$, then*

$$\chi_b'(G \times H) = 2r_G r_H - 1.$$

Proof. Graph $G \times H$ has two components, since both G and H are bipartite. By Lemma 4.2.3 we have $diam(G \times H) \geq 4$ (in every component). Let $g, g' \in V(G)$ be with $d_G(g,g') = k \geq 4$ and $\Delta(G[g]) = r_G - 1 = \Delta(G[g'])$. If k is even, then (g,h) and (g',h) are in the same component of $G \times H$ at distance k, by the distance formula. If k is odd, then (g,h) and (g',h) are in different components, so their distance is ∞. Every induced subgraph of a bipartite graph is also bipartite. Hence $G \times H[(g,h)]$ and $G \times H[(g',h)]$ are bipartite and therefore class 1 graphs by König's Theorem. Moreover $\Delta(G \times H[(g,h)]) = r_G r_H - 1$ and $\Delta(G \times H[(g',h)]) = r_G r_H - 1$ by Lemma 4.2.5. By Theorem 4.2.1 the result follows. □

Notice that all lemmas and both theorems of this section have a symmetric version with respect to the second factor H.

4.3 Computing the b-chromatic index by integer linear programming

By Theorems 4.2.6 and 4.2.7 we are able to tell the exact value of the b-chromatic index for many direct products of regular graphs. It seems that with the growth of $diam(G)$ the chances of $\chi'_b(G)$ being equal to $m'(G)$ do grow as well. In particular, in view of Corollary 4.1.1.1 one would expect for an r-regular graph G that its b-chromatic index is always equal to $m'(G \times C_{2k}) = 2r + 3$ for some relatively small k onwards. (Recall that such a k always exists by the results of [20].) Hence one would "only" need to check some small examples to describe the b-chromatic index of some families of direct products.

Unfortunately, even this can be a difficult, hard work, since up to now no tools have been developed to check small instances (other than brute force). This can be a challenging task for the direct product, which can become quickly a large graph even when the factors considered are small. For this reason we introduce an integer linear programming (ILP) model based on the standard formulation of the vertex coloring problem, to help producing b-colorings of a graph G. Since this is an NP-hard problem, we cannot expect that the solutions obtained by this method lead always to exact values of $\chi'_b(G)$ within reasonable time bounds. Still, every solution produced gives a lower bound for $\chi'_b(G)$. This approach turned out to be quite useful; we present it below.

Let c be an upper bound of $\chi'_b(G)$, and n, m be the number of vertices and edges of G, respectively. We consider the following binary variables:

- for every color $j \in \{1, \dots, c\}$:

$$w_j = \begin{cases} 1 : \text{if color } j \text{ was assigned to some edge} \\ 0 : \text{otherwise} \end{cases} ;$$

- for every edge $e \in \{1, \dots, m\}$ and for every color $j \in \{1, \dots, c\}$:

$$x_{ej} = \begin{cases} 1 : \text{if color } j \text{ is assigned to edge } e \\ 0 : \text{otherwise} \end{cases} ;$$

- for every edge $e \in \{1, \dots, m\}$ and for every color $\in \{1, \dots, c\}$:

$$z_{ej} = \begin{cases} 1 : \text{if edge } e \text{ is dominant of color } j \\ 0 : \text{otherwise} \end{cases} .$$

With these variables we introduce the following ILP model:

maximize $\sum_{j=1}^{c} w_j$

subject to:

- every edge receives exactly one color

  $\sum_{j=1}^{m} x_{ej} = 1$ for every edge $e \in \{1, \ldots, m\}$;

- two incident edges do not get the same color

  $x_{e_1 j} + x_{e_2 j} \leq 1$ for every pair of incident edges e_1 and e_2 and for every color $j \in \{1, \ldots, c\}$;

- w_j has to be 1 if and only if color j was assigned to any edge

  $x_{ej} \leq w_j$ for every edge $e \in \{1, \ldots, m\}$ and for every color $j \in \{1, \ldots, c\}$

  $\sum_{e=1}^{m} x_{ej} \geq w_j$ for every color $j \in \{1, \ldots, c\}$.

For every color, there must be a dominant edge. We accomplish this with help of the next three constraints:

- for every edge e, z_{ej} is equal to 1 if color j is indeed used and there is an edge incident to e of every other color.

  $z_{ej} \leq 1 - (w_{j_1} - \sum_{e_1 \in N(e)} x_{e_1 j_1})$ for every edge $e \in \{1, \ldots, m\}$, and every pair of colors $j, j_1 \in \{1, \ldots, c\}$, $j_1 \neq j$;

- there is a dominant edge for every color used

  $\sum_{e=1}^{m} z_{ej} \geq w_j$ for every color $j \in \{1, \ldots, c\}$;

- edge e can only be dominant of color j if j is assigned to it

  $z_{ej} \leq x_{ej}$ for every edge $e \in \{1, \ldots, m\}$ and for every color $j \in \{1, \ldots, c\}$.

Further, the following classic constraint was used to reduce symmetry:

n	m	density(%)	$\Delta(G)$	$m(G)$	ILP solution
7	4	20	3	3	**3**
7	8	40	3	4	**4**
7	12	60	5	6	**6**
7	16	80	6	8	7
7	21	100	6	11	9
9	7	20	3	3	**3**
9	14	40	6	6	**6**
9	21	60	6	9	8
9	28	80	8	12	10
9	36	100	8	15	12
12	13	20	5	5	**5**
12	26	40	6	10	9
12	39	60	10	14	12
12	52	80	10	18	12
12	66	100	11	21	11
15	21	20	5	7	**7**
15	42	40	10	13	12
15	63	60	11	18	14
15	84	80	14	23	16
15	105	100	14	27	17
30	87	20	11	15	14
30	174	40	17	28	17
30	261	60	22	38	22
30	348	80	26	49	26
30	435	100	29	57	29

Table 4.1: Results for the b-edge chromatic number for some graphs.

- use all colors sequentially

$$w_j \leq w_{j_1} \qquad \text{for all colors}\, j, j_1 \in \{1, \ldots, c\}\, , j_1 < j.$$

It is easy to see from the above constraints that the model produces a b-edge coloring for a graph G (and thus a lower bound for $\chi'_b(G)$), and that a b-edge coloring realizing $\chi'_b(G)$ is obtained when the objective function is maximized.

We have made extensive computational tests with the Cplex technology for linear programming. The model was able to produce optimal solutions only for small/sparse instances of the problem, but achieved nevertheless solutions for all instances within reasonable time limits. We have also run tests on families of graphs for which its b-chromatic index is known from [58] and the obtained results where very accurate.

For the purpose of the computational experience, we use a family of randomly generated graphs with different number of vertices and edge densities. In Table 4.1 we show:

- the number of vertices
- the number of edges
- the percentage of edge's density
- the trivial upper and lower bounds
- the solution obtained by ILP.

Boldface in the ILP solution column indicates that the b-chromatic index was obtained since we got the same value by ILP as is $m'(G)$. Notice that also other values may equal to $\chi'_b(G)$ since $m'(G)$ is an upper bound. In particular for density 100% we have a complete graph K_n and for them it holds that $\chi'_b(G) < m'(G)$, as we know from [58]. Also recall that if a regular graph G has enough vertices, then $\chi'_b(G) = m'(G)$, which follows from the vertex version of the coloring in [20]. But with the increase of the number of vertices of G its density decreases as well. Hence we can expect lower values than $m'(G)$ for less dense graphs. The same can be deduced from the computer experiments of an evolutionary algorithm for $\chi_b(G)$ in [37].

The integer linear integer problem was solved using CPLEX 12.5, running on a Pentium i5 processor, with a 64 bit operating system and 4Gb of available memory.

4.4 Direct products of special graph classes

Recall that for a bipartite graph G we have $G \times K_2 \cong 2G$ [41]. Hence we immediately get the following

Corollary 4.4.0.1. *For a bipartite graph G with $\chi'_b(G) = 2\Delta(G) - 1$ we have*

$$\chi'_b(G \times K_2) = \chi'_b(G).$$

Notice that the condition $\chi'_b(G) = 2\Delta(G) - 1$ cannot be avoided in the above corollary. Namely, graph $2G$ has twice more edges of high degree than G and this means that we can have $m'(2G) > m'(G)$, which could also lead to $\chi'_b(2G) > \chi'_b(G)$. The smallest example for this is $G \cong P_5$, where $m'(P_5) = \chi'_b(P_5) = 2$, but $m'(2P_5) = \chi'_b(2P_5) = 3$.

Next we consider the product of two paths, where a similar situation may arise. Indeed, it could happen that $\chi'_b(P_m \times P_n)$ is strictly greater than the b-chromatic index of one or both of its components. See for instance $P_3 \times P_3$ or $P_4 \times P_6$, respectively.

Theorem 4.4.1. *For any integers $n \geq m \geq 3$ we have*

$$\chi'_b(P_m \times P_n) = \begin{cases} 4 & : \quad m = 3, n < 6 \qquad (1) \\ 5 & : \quad m = 3, n \geq 6 \qquad (2) \\ 5 & : \quad m = 4, n < 6 \qquad (3) \\ 6 & : \quad m = 4, n = 6 \qquad (4) \\ 7 & : \quad m = 4, n > 6 \qquad (5) \\ 6 & : \quad m = 5, n = 5 \qquad (6) \\ 7 & : \quad m \geq 5, n > 5 \qquad (7) \end{cases}$$

Proof. It is easy to see that $m'(P_m \times P_n)$ equals to the expression for $\chi'_b(P_m \times P_n)$ as stated above, with the only exception of $P_3 \times P_5$, where we have $m'(P_3 \times P_5) = 5$. For this case, notice that one component of $P_3 \times P_5$ is isomorphic to two fourcycles which share a common vertex; while the other component is isomorphic to a fourcycle in which each of two fixed opposite vertices have two additional leaves attached. It is easy to see that in every component we can have at most two color dominating edges if we try to find a 5-b-coloring of $P_3 \times P_5$. This yields altogether 4 color dominating edges, which is a contradiction in a 5-b-coloring. Hence $\chi'_b(P_3 \times P_5) < 5$. The upper bound is now clear for all cases.

With help of the ILP model (but it is also easy to verify by hand) we obtained the same values for $\chi'_b(P_m \times P_n)$ in cases (1), (3), (4) and (6), which settle them. For case (2) we did this only for the smallest representative, namely $P_3 \times P_6$. Again we obtained $m'(P_3 \times P_6) = 5 = \chi'_b(P_3 \times P_6)$ computationally (and again it is not hard to do so by hand). Now if n is greater than 6, we can color any subgraph $P_3 \times P_6$ of $P_3 \times P_n$ with the same coloring we obtained. The remaining edges can be colored by the greedy algorithm since degrees of all edges are strictly less than $m'(P_3 \times P_6)$. The remaining cases (5) and (7) are analogous to (2), with representatives $P_4 \times P_7$ and $P_5 \times P_6$, respectively. □

Next we concentrate on the direct product of two cycles.

Theorem 4.4.2. *For any integers $m \geq n \geq 3$ we have*

$$\chi'_b(C_m \times C_n) = \begin{cases} 6 & : \quad (m,n) = (3,3) \\ 7 & : \quad \text{otherwise} \end{cases}$$

Proof. If both m and n are even and $m \geq 8$, then $diam(C_m) \geq 4$ and by Theorem 4.2.7 $\chi'_b(C_m \times C_n) = 7$. If one or both of them are odd different that 3, then $\chi'_b(C_m \times C_n) = 7$ by Theorem 4.2.6. The remaining cases are

$$\{m, n\} \in \{\{3,3\}, \{4,3\}, \{5,3\}, \{6,3\}, \{7,3\}, \{4,4\}, \{6,4\}, \{5,5\}, \{6,6\}\}.$$

We have run them on the ILP model introduced in the previous section and obtained 7-b-colorings for all of them with exception of $C_3 \times C_3$ for which we got a 6-b-coloring. Since $m'(C_m \times C_n) = 7$ we only need to show that there exists no 7-b-coloring of $C_3 \times C_3$.

In order to obtain a contradiction we may assume that $\chi'_b(C_3 \times C_3) = 7$. By Lemma 4.1 from [58], at most two edges of any C_4 or any C_3 in $C_3 \times C_3$ can be color dominating in a 7-b-edge coloring. We will use this result to analyze the subgraph H of $C_3 \times C_3$ induced by seven color dominating edges. Suppose that H contains a vertex v of degree 4. By symmetry, v could be any vertex of $C_3 \times C_3$, and all edges incident to it would be in H. Notice that all but four edges of $C_3 \times C_3$ lie on a C_3 or C_4 with two of its edges incident to v. By the above mentioned Lemma 4.1, only these four edges would be allowed to be color dominating in $C_3 \times C_3$. However, the four edges induce in turn a C_4, so by the same lemma again, only two of them can be color dominating. Thus, only six color dominating edges are possible, this is a contradiction with $C_3 \times C_3$ having a 7-b-edge coloring. So H does not contain a vertex of degree 4.

By a tedious case analysis one can show that H cannot contain a vertex of degree 3 or 2, either. This leads to the conclusion that no 7-b-edge coloring can exist for $C_3 \times C_3$. The details are left to the reader. □

4.5 Resumen del capítulo

Continuamos en este capítulo con el estudio de la versión sobre aristas del problema de b-coloreo, introducida por Jakovac y Peterin en [58]. Un *b-coloreo de aristas* de un grafo G es un coloreo válido de aristas de G tal que cada clase color contiene una arista incidente a por lo menos una arista en cada una de las demás clases color. El *índice b-cromático* de un grafo G es el entero más grande $\chi'_b(G)$ para el cual G tiene un b-coloreo de aristas con $\chi'_b(G)$ colores. Decimos que un b-coloreo de aristas con $\chi'_b(G)$ colores *realiza* $\chi'_b(G)$.

Intuitivamente, para que un b-coloreo de aristas sea posible en un grafo G, se necesitan suficientes aristas de grado lo suficientemente grande, por lo menos una para cada clase color. Sea $e_1, \ldots, e_n$ una secuencia de aristas, tal que $d(e_1) \geq \cdots \geq d(e_n)$, donde $d(e_i)$ denota el grado de e_i. Entonces $m'(G) = max\{i : d(e_i) \geq i - 1\}$ es una cota superior para $\chi'_b(G)$. En [58], los autores determinaron el índice b-cromático para árboles, y dieron condiciones para grafos que poseen índice b-cromático estrictamente menor a $m'(G)$, así como condiciones para las cuales $\chi'_b(G) = m'(G)$. Demostraron además que $\chi'_b(G) = 5$ para grafos cúbicos conexos, con únicamente cuatro excepciones: K_4, $K_{3,3}$, el prisma sobre K_3, y el cubo Q_3. El problema de computar el índice b-cromático es NP-completo; este resultado fue mostrado por Lima et al. en [78].

El *producto directo* $G \times H$ de grafos G y H tiene conjunto de vértices $V(G) \times V(H)$; dos vértices (g, h) y (g', h') son adyacentes en $G \times H$ si son adyacentes en ambas coordenadas, i.e. $gg' \in E(G)$ y $hh' \in E(H)$. Si $e = (g, h)(g', h') \in E(G \times H)$, sea $p_G(e) = gg'$ and $p_H(e) = hh'$ la *proyección de una arista* e sobre G y H, respectivamente. El producto directo parece ser el más difícil de trabajar entre los cuatro productos estándar (Cartesiano, strong, directo and lexicográfico). La razón de esto es el hecho que cada arista de $G \times H$ proyecta a una arista en ambos factores, propiedad que no se verifica en los demás productos. Aún propiedades básicas de grafos como la conexidad son no triviales para el producto directo. Por ejemplo, $G \times H$ no necesariamente es conexo, aún si ambos factores lo son. Se puede demostrar que si ambos factores son bipartitos y conexos, el producto directo tiene exactamente dos componentes conexas (see [98]). Desde su formulación inicial por Weichsel [98] en 1962, el producto directo de grafos fue intensamente estudiado en las áreas de coloreo de grafos, reconocimiento y descomposición de grafos, embedding de grafos, teoría de matching y estabilidad en grafos (ver el libro [42] para más detalles). Existe un problema, sin embargo, que es considerado el principal interrogante abierto para el producto directo, y está relacionada con coloreo de grafos. En 1966, Hedetniemi [49] conjeturó que para todo par de grafos G y H, $\chi(G \times H) = min\{\chi(G), \chi(H)\}$. Esta famosa conjetura ha resistido los esfuerzos de los investigadores hasta el día de hoy.

En este capítulo, describimos una nueva cota para el índice b-cromático del producto de un grafo G y un grafo regular H que admite una partición de sus aristas en matchings perfectos (esta partición se denomina *one-factorization*).

Teorema: Sea G un grafo y H un grafo r-regular. Si H tiene una one-factorization, entonces

$$\chi'_b(G \times H) \geq r\chi'_b(G).$$

Además, presentamos resultados exactos para χ'_b para el producto directo de algunos grafos conexos regulares.

Teorema: Sean G y H grafos conexos, r_G- y r_H-regulares, respectivamente, y sin triángulos. Adicionalmente, sea $gg' \in E(G)$, donde g y g' no se encuentran en ningún ciclo de 5 vértices y $\Delta(G[g]) = r_G - 1 = \Delta(G[g'])$. Si por lo menos uno de los grafos G o H es no bipartito, entonces

$$\chi'_b(G \times H) = 2r_G r_H - 1.$$

Teorema: Sean G y H grafos bipartitos conexos, y r_G- y r_H-regulares, respectivamente. Sea $diam(G) \geq 4$. Si existe $g, g' \in V(G)$ con $d_G(g, g') \geq 4$ y $\Delta(G[g]) = r_G - 1 = \Delta(G[g'])$, entonces

$$\chi'_b(G \times H) = 2r_G r_H - 1.$$

Determinar el índice b-cromático de un grafo puede ser muy tedioso, aún para ejemplos pequeños. Por esta razón, desarrollamos también un sencillo modelo de programación lineal entera para el problema. Con este modelo y los resultados anteriores, nos fue posible determinar valores exactos para χ'_b para el producto directo de caminos y el producto directo de ciclos.

Teorema: Para dos enteros $n \geq m \geq 3$, se tiene

$$\chi'_b(P_m \times P_n) = \begin{cases} 4 & : \quad m = 3, n < 6 \qquad (1) \\ 5 & : \quad m = 3, n \geq 6 \qquad (2) \\ 5 & : \quad m = 4, n < 6 \qquad (3) \\ 6 & : \quad m = 4, n = 6 \qquad (4) \\ 7 & : \quad m = 4, n > 6 \qquad (5) \\ 6 & : \quad m = 5, n = 5 \qquad (6) \\ 7 & : \quad m \geq 5, n > 5 \qquad (7) \end{cases}$$

Teorema: Para dos enteros $m \geq n \geq 3$, se tiene

$$\chi'_b(C_m \times C_n) = \begin{cases} 6 & : \quad (m, n) = (3, 3) \\ 7 & : \quad \text{en otro caso} \end{cases}$$

Los resultados de este trabajo fueron propuestos en [74].

CHAPTER 5

A general cut-generating procedure for the stable set polytope

5.1 Introduction

Given an undirected graph G, the *maximum cardinality stable set problem* (MSS) asks for a stable set S in G of maximum cardinality. MSS is NP-hard, and has been approached in the literature through several techniques. A number of exact methods have been developed to solve it, see [14] for a survey.

Although combinatorial methods for MSS (like those in [91, 94]) perform better than branch and cut algorithms, it is of great interest to continue the search for efficient polyhedral methods for this problem. This is due to the facts that (a) MSS frequently appears as a sub-structure in many combinatorial optimization problems, (b) in many situations MSS is solved as a sub-routine for generating valid inequalities for general mixed integer programs (see, e.g., [3]), and (c) real applications may need specific versions of MSS with additional constraints and in this context integer programming often turns out to be effective.

Mannino and Sassano [82] introduced in 1996 the idea of edge projections as a specialization of Lovász and Plummer's clique projection operation [79]. Many properties of edge projections are discussed in [82] and, based on these properties, a procedure computing an upper bound for MSS is developed. This bound is then incorporated in a branch and bound scheme. Rossi and Smriglio take these ideas into an integer programming environment in [88], where a separation procedure based on edge projection is proposed. This procedure iteratively removes and projects edges with certain properties, and heuristically finds violated rank inequalities (i.e., inequalities of the form $\sum_{v \in A} x_v \leq \alpha(G[A])$, where $A \subseteq V$ and $G[A]$ is the subgraph of G induced by A). Finally, Pardalos et al. [87] extend the theory of edge projection by explaining the facetness properties of the inequalities obtained by this procedure. The authors give a branch and cut algorithm that uses edge projections as a separation tool, as well as several known families of valid inequalities such as the odd hole inequalities (with a polynomial-time exact separation algorithm), the clique inequalities (with heuristics), and mod-$\{2, 3, 5, 7\}$ cuts.

Edge projections are a special case of Lovász and Plummer's clique projections [79]. Rossi and Smriglio propose in [88] to employ a sequence of edge projection operations to reduce the original graph G and make it denser at the same time, allowing for a faster identification of clique inequalities on the reduced graph G'. A key step for achieving this is to be able to establish how $\alpha(G)$ is affected by these edge projections, or, in other words, how exactly $\alpha(G)$ relates to $\alpha(G')$. We aim at generalizing Rossi and Smriglio's procedure by projecting cliques instead of edges, so we also need to show how $\alpha(G)$ changes as a result of this operation. Our method allows thus to establish a more general relation between G and G'.

In this chapter we propose the use of clique projections as a general method for cutting plane generation for the MSS, along with a new clique lifting procedure that leads to stronger inequalities than those obtained with the edge projection method. The proposed method is able to generate both rank and generalized rank valid inequalities (to be defined below), by resorting to the general lifting procedure introduced in [100]. This approach allows to produce cuts of a quite general nature, including cuts from the known families of valid inequalities for the MSS polytope. This approach departs from the usual template-based paradigm for generating cuts, and seeks to unify and generalize the separation procedures for the known cuts. In this sense, our main goal is to provide a more complete understanding of the maximum stable set polytope, which may help also in the solution of other combinatorial optimization problems. In Section 5.2 we define the MSS polytope $STAB(G)$ and state some useful properties. Section 5.3 defines the operation of clique projection and explores some basic facts on this operation. In Sections 5.4 and 5.5 we introduce our cut-generating method, by applying the lifting method presented in [100]. Finally, in Section 5.6 we present some preliminary computational experience on the DIMACS instances, which show that the method is competitive. These results were submitted for publication in [29].

5.2 The maximum stable set polytope

Let $n = |V|$ and $\mathcal{S}(G) \subseteq \{0, 1\}^n$ be the set of all characteristic vectors of stable sets of G. We will write simply $\mathcal{S}$ when G is clear from context. For $W \subseteq V$, $\mathcal{S}(G[W])$ stands for the characteristic vectors of stable sets of G involving vertices in W only. The polytope of stable sets of G is denoted by

$$STAB(G) = \text{conv}\{x \mid x \in \mathcal{S}(G)\}.$$

Note that the stability number of G is $\alpha(G) = max\{\sum_{v \in V} x_v \mid x \in STAB(G)\}$. If $c \in \mathbb{R}^n$, then the *weighted stable number of G, according to c* is $\alpha(G, c) = max\{c^\top x \mid$

$x \in STAB(G)\}$. The general form of a facet-inducing inequality of $STAB(G)$ is

$$c^\top x \leq \alpha(G[H], c), \tag{5.1}$$

where $c \in \mathbb{R}^n$, $c \geq \mathbf{0}$, and $H = \{v \in V \mid c_v > 0\}$. Note that if $c \in \{0,1\}^n$ then we have the rank inequality mentioned in the Introduction.

The following is an upper bound for the weighted stable set number based on the unweighted one.

Lemma 5.2.1. *Let $c \in \mathbb{R}^n$, $c \geq \mathbf{0}$, and $c_{min} = min\{c_v \mid v \in V, c_v > 0\}$. If $\bar{c} \in \mathbb{R}^n$ is such that $\bar{c}_v = 0$, if $c_v = 0$, and $\bar{c}_v = c_v - c_{min}$, otherwise, then $\alpha(G,c) \leq c_{min}\alpha(G) + \alpha(G,\bar{c})$.*

Proof. The weight of a stable set S containing exactly s nonnull weight vertices can be written as

$$\sum_{v \in S} c_v = sc_{min} + \sum_{v \in S}(c_v - c_{min}) \leq sc_{min} + \alpha(G,\bar{c}).$$

The result follows from $s \leq \alpha(G)$. □

Suppose we have a heuristic H for computing $\alpha(G)$. With the help of Lemma 5.2.1 we can obtain a simple heuristic for the weighted maximum stable set problem $\alpha(G,c)$ as follows. Subtract in step j the minimum element $c_{min_{j-1}}$ from every coefficient of vector $\bar{c}_{j-1}$ ($\bar{c}_0 = c$), as in Lemma 5.2.1. Perform this operation until (say after k steps) the number of non null elements remaining in vector $\bar{c}_k$ allows for exact enumeration of $\alpha(G,\bar{c}_k)$. Then an upper bound for $\alpha(G,c)$ is $\alpha(G)(\sum_{1 \leq j < k} c_{min_j}) + \alpha(G,\bar{c}_k)$

We present this lemma here since it will be useful for our clique-lifting operation in Section 5.4. This operation involves the problem of finding an upper bound for the maximum weight of a stable set in a subgraph of G.

5.3 Clique projection

The edge projection operation as defined by Rossi and Smriglio involves the removal of vertices in the common neighborhood of the endpoints of the edge being projected, whereas the clique projection operation defined here does not remove these vertices. Due to this fact, the clique projection defined here does not correspond to the standard edge projection when the projected clique is an edge. The motivation for this variation in the definition will become clear in the remainder of this chapter. Define $N_W = \cap_{w \in W} N(w)$ and $N_{uv} = N(u) \cap N(v)$.

Definition 5.3.1 (Clique projection [79])**.** *Let $W \subseteq V$, $|W| \geq 2$, be a clique in G. The* clique projection *of W gives the graph $G \mid W = (V, E \mid W)$ in which $E \mid W = E \cup \{xy \notin E \mid W \subseteq N(x) \cup N(y)\}$.*

The edges in $(E \mid W)\backslash E$ (i.e., the added edges after the projection) are called *false edges*. If $W = \{u, v\}$ for some $uv \in E$ and we remove the vertices in $N_{uv} \cup \{u, v\}$ when performing the projection, we have the edge projection explored in [87, 88].

Definition 5.3.2 ([88])**.** *An edge $uv \in E$ is* projectable *in G if and only if there exists a maximum stable set S in G such that $S \cap \{u, v\} \neq \emptyset$.*

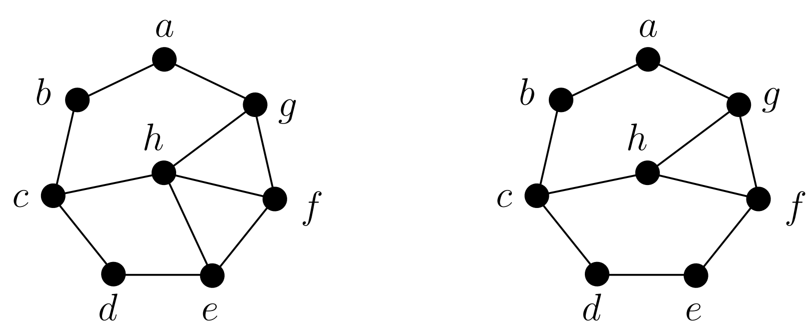
Lemma 5.3.1 ([82])**.** *If $uv \in E$ is a projectable edge in G, then $\alpha(G) = \alpha(H[V \setminus (N_{uv} \cup \{u, v\})]) + 1$, where $H = G \mid \{u, v\}$.*

Definition 5.3.3. *A clique $W \subseteq V$, $|W| \geq 2$, is* projectable *in G if and only if there exists a maximum stable set S in G such that $S \cap W \neq \emptyset$.*

Lemma 5.3.2. *If a clique $W \subseteq V$, $|W| \geq 2$, is projectable in G, then $\alpha(G) = \alpha(H[V \setminus (N_W \cup W)])$ or $\alpha(G) = \alpha(H[V \setminus (N_W \cup W)]) + 1$, where $H = G \mid W$.*

Proof. $\alpha(H[V \setminus (N_W \cup W)]) \leq \alpha(G[V \setminus (N_W \cup W)])$ is a direct consequence of $E \subseteq E \mid W$. Thus, $\alpha(G)$ is an upper bound for $\alpha(H)$. On the other hand, W being projectable and $E \mid W \subseteq E \mid \{u, v\}$, for any edge uv such that $u, v \in W$, imply $\alpha(H[V \setminus (N_W \cup W)]) \geq \alpha(G) - 1$ by Lemma 5.3.1, giving the desired lower bound. □

Both cases of the above lemma may happen, as illustrated in the examples in Figure 5.1.



(a) $\alpha(G) = 3$ and $\alpha(H[V \setminus (N_W \cup W)]) = 2$, where $H = G \mid W$.

(b) $\alpha(G) = \alpha(H[V \setminus (N_W \cup W)]) = 3$, where $H = G \mid W$.

Figure 5.1: Examples of the two cases of Lemma 5.3.2 with $W = \{f, g, h\}$. In (a), $\alpha(G) = \alpha(H[V \setminus (N_W \cup W)]) + 1$, whereas $\alpha(G) = \alpha(H[V \setminus (N_W \cup W)])$ in (b), where $H = G \mid W$.

Corollary 5.3.2.1. *If a clique $W \subseteq V$, $|W| \geq 2$, is projectable in G, then $\alpha(G[V \setminus (N_W \cup W)]) = \alpha(H[V \setminus (N_W \cup W)])$ or $\alpha(G[V \setminus (N_W \cup W)]) = \alpha(H[V \setminus (N_W \cup W)]) + 1$, where $H = G \mid W$.*

Proof. The upper bound $\alpha(H[V \setminus (N_W \cup W)]) \leq \alpha(G[V \setminus (N_W \cup W)])$ is a direct consequence of $E \subseteq E \mid W$. For the lower bound, we use $\alpha(G[V \setminus (N_W \cup W)]) \leq \alpha(G)$ and Lemma 5.3.2 (since W is projectable) to write $\alpha(G[V \setminus (N_W \cup W)]) \leq \alpha(H[V \setminus (N_W \cup W)]) + 1$. □

5.4 Clique-Lifting

In this section we lay the basic facts for the cut-generating procedure for the MSS polytope. In particular, we are interested in applying the lifting procedure presented by Xavier and Campêlo in [100], which is our main tool. We present this procedure and show how it can be applied in the particular MSS setting for the well-known clique inequalities.

We start with some preliminary definitions. Given a valid inequality

$$\sum_{v \in W} \pi_v x_v \leq \beta \tag{5.2}$$

for $STAB(G)$ with $W \subseteq V$, $\beta \in \mathbb{R}$, and $\pi_v \neq 0$ for all $v \in W$, we say that W is the *support of* (5.2) and we denote by $F_W(\beta, \pi_{v\in W}) = \{x \in STAB(G) \mid \sum_{v\in W} \pi_v x_v = \beta\}$ the face induced by W, β, and π, in $STAB(G)$.

Lemma 5.4.1 (Lifting lemma [100])**.** *Let $U \subseteq V$, $\beta \in \mathbb{R}$ and $\pi_v \in \mathbb{R}$, for all $v \in U$, such that $\sum_{v\in U} \pi_v x_v \leq \beta$ is a valid inequality for $STAB(G)$. If $c^\top x - d \leq 0$, $c, x \in \mathbb{R}^n$ and $d \in \mathbb{R}$, is a valid inequality for $F_U(\beta, \pi_{v\in U})$, then*

$$L_{x,\lambda}(U) = (c^\top x - d) - \lambda \left(\sum_{v \in U} \pi_v x_v - \beta \right) \leq 0, \tag{5.3}$$

with

$$\lambda \leq \min \left\{ \frac{c^\top x - d}{\sum_{v\in U} \pi_v x_v - \beta} \mid x \in \mathcal{S}, \sum_{v \in U} \pi_v x_v < \beta \right\}, \tag{5.4}$$

is a valid inequality for $STAB(G)$.

A value λ that satisfies (5.4) is said to be *valid* for a lifting of $c^\top x - d \leq 0$ with respect to U. Since λ appears with a negative sign in (5.3), it turns out that if λ_1 and λ_2 are valid and $\lambda_1 < \lambda_2$, then

$$\{x \notin STAB(G) \mid L_{x,\lambda_1}(U) \leq 0\} \subset \{x \notin STAB(G) \mid L_{x,\lambda_2}(U) \leq 0\}.$$

Consequently, the greater the coefficient λ is, the stronger the inequality (5.3) becomes. Sufficient conditions for (5.3) to be facet-defining for $STAB(G)$ are stated next.

Theorem 5.4.2 ([100])**.** *If $F_U(\beta, \pi_{v\in U})$ is a facet of $STAB(G)$, $c^\top x \leq d$ is facet-defining for $F_U(\beta, \pi_{v\in U})$ and λ satisfies* (5.4) *at equality, then* (5.3) *is facet-defining for $STAB(G)$.*

Define $\mathbf{1}_{v\in W}$ as the binary size-n vector y such that $y[v] = 1$ if and only if $v \in W$. A special case of Lemma 5.4.1 occurs when U is a clique in G such that $U = W \cup N_W$. In such a case, if $c^\top x - d \leq 0$ is a valid inequality for $F_{W\cup N_W}(1, \mathbf{1}_{v\in W\cup N_W})$, then the application of the lifting operation (5.3), for any valid λ, is called *clique-lifting of $c^\top x - d \leq 0$ with respect to W*. The following lemma establishes a sufficient condition for a clique-lifting operation to result in a valid inequality for $STAB(G)$.

Lemma 5.4.3. *Let $W \subseteq V$, $|W| \geq 2$, be a clique in G and $c^\top x - d \leq 0$, $c, x \in \mathbb{R}^n$ and $d \in \mathbb{R}$, be a valid inequality for $STAB(G \mid W)$. If W contains a vertex w such that $N(w) \setminus W$ is a clique in G, then $c^\top x - d \leq 0$ is also valid for $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W})$.*

Proof. If $E \mid W = \emptyset$, then there is nothing to prove since $STAB(G \mid W) = STAB(G)$ and $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W}) \subseteq STAB(G \mid W)$ in this case. Otherwise, let $x \in F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W}) \cap \mathcal{S}$ (x is an integer point in $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W})$) and uv be a false edge in $G \mid W$. By definition of x, there exists $z \in W \cup N_W$ such that $x_z = 1$. We consider two cases to show that $\{uz, vz\} \cap E \neq \emptyset$. First, if $z \in W$, then the claim holds because $W \subseteq N(u) \cup N(v)$ by definition of clique-projection. Second, $z \in N_W$ and, by hypothesis, let $w \in W$ be such that $N(w) \setminus W$ is a clique in G. Since $\{u, v\} \cap N(w) \neq \emptyset$ (again because $W \subseteq N(u) \cup N(v)$), the claim follows. We conclude that $x_u = 0$ or $x_v = 0$ and, consequently, x corresponds to a stable set of $G \mid W$, which means that $c^\top x - d \leq 0$ holds. The lemma stems from the fact that $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W})$ is a convex hull of the points in $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W}) \cap \mathcal{S}$. □

A clique-lifting operation involves the problem of finding an upper bound for the maximum weight of a stable set in a subgraph of G. The following lemma establishes one such upper bound based on Lemma 5.2.1.

Lemma 5.4.4. *Let $W \subseteq V$, $|W| \geq 2$, be a clique in G and $c^\top x - d \leq 0$, $c, x \in \mathbb{R}^n$ and $d \in \mathbb{R}$, be a valid inequality for $STAB(G \mid W)$ with support H. Moreover, let $c_{min} = \min\{c_v \mid v \in H \setminus (W \cup N_W), c_v > 0\}$. If W contains a vertex w such that $N(w) \setminus W$ is a clique in G, then*

$$\lambda = d - c_{min}\alpha(G[H \setminus (W \cup N_W)]) - \max_{x \in \mathcal{S}(G[H \setminus (W \cup N_W)])} \sum_{v \in H \setminus (W \cup N_W)} \max\{0, c_v - c_{min}\} x_v \quad (5.5)$$

is valid for the clique-lifting of $c^\top x - d \leq 0$ with respect to W.

Proof. Lemma 5.4.1, combined with Lemma 5.4.3, establishes that λ is valid for the clique-lifting of $c^\top x - d \leq 0$ with respect to W if

$$\begin{aligned} \lambda &\leq \min\left\{ \frac{c^\top x - d}{\sum_{v \in W \cup N_W} x_v - 1} \mid x \in \mathcal{S}, \sum_{v \in W \cup N_W} x_v = 0 \right\} \\ &= \min\left\{ -c^\top x + d \mid x \in \mathcal{S}(G[H \setminus (W \cup N_W)]) \right\}, \end{aligned}$$

where H is the support of $c^\top x - d \leq 0$, which means that

$$\lambda \leq d - \max\left\{ c^\top x \mid x \in \mathcal{S}(G[H \setminus (W \cup N_W)]) \right\}. \quad (5.6)$$

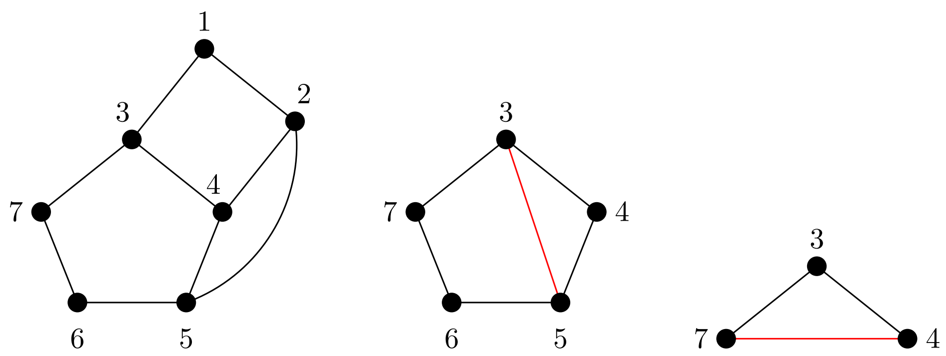
Lemma 5.2.1 implies that the last two terms of the righthand side of (5.5) provide an upper bound for the maximum weight of a stable set of $G[H \setminus (W \cup N_W)]$. Thus, the result holds due to (5.6). □

The following result is a particular case of Lemma 5.4.4 and the basis of the separation procedure proposed in [88].

Corollary 5.4.4.1. *Let* $H \subseteq V$*. If* $W \subseteq V$ *is projectable in* $G[H \cup W \cup N_W]$ *and* $c^\top x - d \leq 0$ *is the rank inequality for the subgraph of* $G \mid W$ *induced by* $H \cup W \cup N_W$*, then* $\lambda = -1$ *is valid for the clique-lifting of* $c^\top x - d \leq 0$*.*

Proof. Using (5.5) with $c_v = 1$, for all $v \in H \setminus (W \cup N_W)$, and $\alpha(G[H \setminus (W \cup N_W)]) \leq \alpha(G \mid W[H \setminus (W \cup N_W)]) + 1$. □

A remark in connection with this result is that $\alpha(G[H]) = \alpha(G \mid W[H])$ implies that $\lambda = 0$ is valid for the clique-lifting of $c^\top x - d \leq 0$. Such a situation is depicted in Figure 5.2.



(a) Graph $G = G^{(0)}$, $\alpha(G) = 3$. It generates the valid inequality $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \leq 3$, which is the summation of the rank inequality $x_3 + x_4 + x_5 + x_6 + x_7 \leq 2$ with the clique one $x_1 + x_2 \leq 1$.

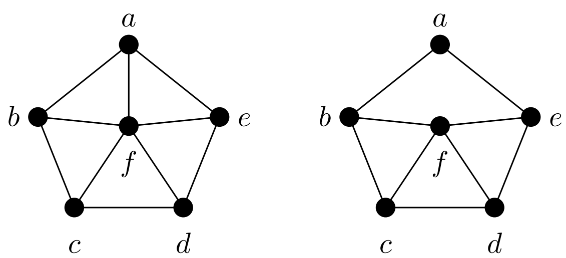
(b) Graph $G^{(1)}$, $\alpha(G^{(1)}) = 2$, after projecting edge $(1, 2)$ containing one false edge $((3, 5))$. It generates the inequality $x_3 + x_4 + x_5 + x_6 + x_7 \leq 2$, which is valid for G.

(c) Graph $G^{(2)}$, $\alpha(G^{(2)}) = 1$, after projecting edge $(5, 6)$ containing one false edge $((4, 7))$. It generates the inequality $x_3 + x_4 + x_7 \leq 1$, which is not valid for $G^{(1)}$.

Figure 5.2: Example of a sequence of edge projections such that the final lifted inequality is weaker than an intermediary one. Deleting false edges from $G^{(1)}$ does not increase the maximum size of a stable set.

5.5 The cut-separating procedure

Successive applications of the clique projection operation and the corresponding clique-lifting operations according to Lemma 5.4.3, lead to stronger inequalities than those that can be obtained with the edge projection method proposed in [88]. In fact, the edge projection corresponds to a special case of Lemma 5.4.3 in which inequality $c^\top x - d \leq 0$ is a rank inequality of a projected graph's clique with empty intersection with W. As an illustration, consider the structure in Figure 5.3(a) and $W = \{d, e, f\}$. The de projection in this graph, followed by the antiprojection of the clique $\{a, b, c\}$ of $G \mid de$, gives the rank inequality $x_a+x_b+x_c+x_d+x_e+x_f \leq 2$. The same inequality is obtained with Lemma 5.4.3 if we take as $c^\top x - d \leq 0$ the clique inequality of $G+de$ for $\{a,b,c\}$. Nevertheless, even in this simple example, there is an inequality that cannot be derived with the method in [88]. If we take $\{a,b,c,f\}$ as the clique inducing set of vertices associated with $c^\top x - d \leq 0$ in Lemma 5.4.3, then we get $x_a+x_b+x_c+x_d+x_e+2x_f \leq 2$ as a valid (indeed, facet-defining [21]) inequality for $STAB(G)$.



(a) Not rank inequality. (b) Rank inequality.

Figure 5.3: Structures leading to stronger inequalities than edge projection.

The structure in Figure 5.3(b) (assuming that it induces a rank inequality of G [8, 21]) also illustrates the fact that $\sum_{v \in W} x_v \leq 1$ being facet-defining for $STAB(G)$ is not necessary to derive another facet of $STAB(G)$. To show this, we choose $W = \{d,e\}$ and still take the clique inequality of $G+de$ associated with $\{a,b,c,f\}$. With such a configuration, Lemma 5.4.3 gives the rank inequality $x_a+x_b+x_c+x_d+x_e+x_f \leq 2$ as well. Observe that this inequality is not derived by the method in [88] if edge ae is deleted before projecting de ($x_b+x_c+x_d+x_e+x_f \leq 2$ would be generated instead).

We are now in position of presenting the cut-generating procedure that is the main contribution of this work. Algorithm 2 summarizes the proposed procedure: we generate and project a sequence of cliques until we find a violated clique inequality. At this point, we antiproject the cliques in reverse order and apply the Lemma 5.4.1 in order to get a valid inequality for the original graph.

Let $\langle W^{(0)}, \ldots, W^{(k)} \rangle$ be the sequence of $k \geq 0$ cliques $W^{(i)} \subseteq V$, $|W^{(i)}| \geq 2$, generated by the algorithm. Also, let $G^{(0)} = G, \ldots, G^{(k)}$ be the sequence of projected graphs within the algorithm. For $i \in \{0,\ldots,k\}$, there exists $w^{(i)} \in W^{(i)}$ and $R^{(i)}$ a minimal subset of $N_{G^{(i)}}(w^{(i)})$ such that $N_{G^{(i)}}(w^{(i)}) \setminus (W^{(i)} \cup R^{(i)})$ is a clique in $G^{(i)}$. Put differently, $R^{(i)}$ is a minimal subset of vertices such that the removal of all edges $w^i v$, $v \in R^{(i)}$, results in a graph $\tilde{G}^{(i)}$ in which $N_{\tilde{G}^{(i)}}(w^{(i)}) \setminus W^{(i)}$ is a clique. Observe that this is the sufficient condition stated in Lemma 5.4.4. In addition, $W^{(i)}$ has the following property in $G^{(i)}$.

Lemma 5.5.1. *Let $W \subseteq V$, $|W| \geq 2$, be a clique in G. If there exists $w \in W$ such that $N(w) \setminus W$ is a clique, then W is projectable in every subgraph of G induced by $H \subseteq V$ such that $W \subseteq H$.*

Proof. Let $H \subseteq V$ such that $W \subseteq H$. If S is a maximum stable set of $G[H]$ and

Algorithm 2 Cut-generating procedure

Find a starting clique $W^{(0)}$ of G;
$k := 0$;
while $x(W^{(k)}) \leq 1$ **do**
  Remove edges such that $N(w) \backslash W^{(k)}$ is a clique, for some $w \in W^{(k)}$;
  Project the clique $W^{(k)}$, getting the graph $G^{(k+1)}$;
  Find a clique $W^{(k+1)}$;
  $k := k + 1$;
end while
Let $\pi^{(k)}$ be the characteristic vector of $W^{(k)}$ (so the inequality $\pi^{(k)} x \leq \gamma^{(k)} := 1$ is violated);
for $i \leftarrow k, \ldots, 1$ **do**
  Apply Lemma 5.4.4 to $\pi^{(k)} x \leq \gamma^{(k)}$ and $W^{(k)}$ in graph $G^{(k-1)}$, obtaining a new inequality $\pi^{(k-1)} x \leq \gamma^{(k-1)}$ valid for $G^{(k-1)}$;
end for
return $\pi^{(0)} x \leq \gamma^{(0)}$, if violated;

$S \cap W = \emptyset$, then $|S \cap N(w)| = 1$ since $N(w) \backslash W$ is a clique. Thus, $S' = (S \backslash N(w)) \cup \{w\}$ is a maximum stable set of $G[H]$ such that $W \cap S' \neq \emptyset$. $\square$

Let us assume that $\bar{x} \notin STAB(G^{(k)})$ be such that $\mathbf{1}^\top_{v \in W^{(i)}} \bar{x} \leq 1$, for all $i \in \{0, \ldots, k-1\}$, and $\mathbf{1}^\top_{v \in W^{(k)}} \bar{x} > 1$. Applying Lemma 5.4.3 with $\lambda^{(k)}$ obtained with Lemma 5.4.4, we obtain the inequality

$$\mathbf{1}^\top_{v \in W^{(k-1)}} x - 1 - \lambda^{(k)} (\mathbf{1}^\top_{v \in W^{(k)}} x - 1) \leq 0,$$

valid for $STAB(G^{(k-1)})$. If the subgraph of $G^{(k-1)}$ induced by $W^{(k)} \backslash (W^{(k-1)} \cup N_{W^{(k-1)}})$ does not increase its stability number with respect to $G^{(k)}$ (this happens if, for instance, $G^{(k)}$ has no false edges), then $\lambda^{(k)} = 0$ and $\mathbf{1}^\top_{v \in W^{(k)}} x \leq 1$ is valid for $STAB(G^{(k-1)})$. Otherwise, by the definition of clique-projection, $\alpha(G^{(k-1)}[W^{(k)} \backslash (W^{(k-1)} \cup N_{W^{(k-1)}})]) = 2$ and $\lambda^{(k)} = -1$. Thus, the new inequality becomes

$$\mathbf{1}^\top_{v \in W^{(k-1)} \oplus W^{(k)}} x + \mathbf{2}^\top_{v \in W^{(k-1)} \cap W^{(k)}} x \leq 2,$$

which is a rank inequality only if $W^{(k-1)} \cap W^{(k)} = \emptyset$. Since $\mathbf{1}^\top_{v \in W^{(k)}} \bar{x} - 1 \leq 0$, this new inequality may be violated by $\bar{x}$.

Let us take Figure 5.2 as an example of a sequence of $k = 2$ clique-projections with the point $\bar{x} = \mathbf{1/2}_{v \in \{1,\ldots,7\}}$. The corresponding sequence of cliques is $W^{(0)} = \{1, 2\}$ ($w^{(0)} = 2$), $W^{(1)} = \{5, 6\}$ ($w^{(1)} = 5$), and $W^{(2)} = \{3, 4, 7\}$ ($w^{(2)} = 3$). Inequality $x_3 + x_4 + x_7 \leq 1$ is violated by $\bar{x}$ since $\bar{x}_3 + \bar{x}_4 + \bar{x}_7 = 1.5$. Removing edge $(4, 7)$ from $G^{(2)}$ creates a stable set of size 2, and (5.5) gives $\lambda^{(2)} = -1$, as expected. Thus, (5.3) results in

$$x_3 + x_4 + x_5 + x_6 + x_7 \leq 2, \tag{5.7}$$

which is valid for $STAB(G^{(1)})$ and violated by $\bar{x}$. The clique-lifting of (5.7) with Lemma 5.4.4 gives $\lambda^{(1)} = 0$.

Instances				Root subproblem				
$G = (V, E)$	$\|V\|$	Dens.	$\alpha(G)$	LB	UB	UB[88]	UB[87]	Time
brock200_2	200	0.50	12	10	21.67	22.01	20.99	48.54
brock200_4	200	0.34	17	15	30.69	30.87	29.93	59.56
c_fat500-1	500	0.96	14	14	14.00	14.90	14.00	222.29
c_fat500-2	500	0.93	26	26	29.32	57.78	26.97	$***$
C125.9	125	0.10	34	34	40.90	37.40	41.26	16.36
C250.9	250	0.10	44	43	69.56	58.30	69.76	245.10
keller4	171	0.35	11	11	14.81	14.95	14.83	14.57
san200_0.7_2	200	0.30	18	18	18.00	19.18	18.50	155.74
p_hat300-2	300	0.51	25	25	34.07	34.19	33.81	145.00
p_hat300-3	300	0.26	36	36	54.82	53.19	54.12	$***$
$G(100, 0.10)$	100	0.10	31.2	30.6	34.50	-	-	4.43
$G(100, 0.20)$	100	0.20	20.2	19.6	25.70	-	-	3.75

Table 5.1: Results with graphs selected from the DIMACS benchmark.

Instances	Number of cuts		
$G = (V, E)$	Clique	Rank	Weighted
brock200_2	932	30	26
brock200_4	364	36	8
c_fat500-1	1426	156	9
c_fat500-2	1117	439	13
C125.9	27	194	1
C250.9	92	187	14
keller4	296	101	87
san200_0.7_2	1251	648	136
p_hat300-2	1952	509	71
p_hat300-3	656	149	29
$G(100, 0.10)$	1.6	131.8	2.8
$G(100, 0.20)$	42.4	98.8	20.8

Table 5.2: Results with graphs selected from the DIMACS benchmark (cont.).

In the general case, a nonempty intersection of $W^{(i-1)}$ and $W^{(i)}$ leads to an inequality with coefficients greater than 1. This is the case in the example of Figure 5.3.

5.6 Preliminary computational experiments

In this section we provide preliminary computational experiments in order to explore whether the proposed method is useful as a cut-generating tool or not. Our main goal is not to provide a competitive algorithm for MSS, since combinatorial algorithms are much more effective than cutting-plane algorithms for this problem. Anyway, we intend to assess whether the proposed procedure is effective at generating generalized rank cuts for the MSS polytope, and the nature of the obtained cuts.

To this end, we implemented the cut-generating procedure as a separation procedure attached to Cplex 12.6's branch and cut algorithm to compute a strengthened upper bound for the root subproblem. Whenever a fractional solution is found, we execute the cut-generating procedure several times, each execution starting from a different clique. In order to ensure that the initial cliques are not repeated, we employ a backtracking-based enumeration that in principle enumerates all cliques in the graph. We employ for this purpose Östergaard's algorithm in [86]. We do not generate all cliques; instead, we stop when a prespecified number of initial cliques is found.

For each initial clique, we generate a sequence of cliques with a greedy algorithm that tries not to repeat too many vertices already belonging to a clique in the sequence. We project each clique in the sequence, until a simple greedy heuristic finds a violated clique inequality. When this happens, we apply the clique-lifting procedure and check if the generated inequality (which is guaranteed to be valid) is violated.

The initial model contains only the constraint $x_i + x_j \leq 1$ for every edge $ij \in E$. We implement a simple greedy heuristic for calculating a lower bound at the root node in the enumeration tree. We also implement the heuristic in Lemma 5.2.1 for calculating a dual bound for λ within the cut-generating procedure. We first search for rank inequalities (by projecting at each step a clique disjoint to the preceding cliques), which include clique inequalities. If such inequalities are not found, then we allow for projecting cliques with nonempty intersection with the preceding cliques, hence generalized rank inequalities can be obtained in this case. The size $s \in \mathbb{Z}_+$ of the analyzed cliques is a parameter of our implementation, and we try with $2 \leq s \leq 7$ in the experiments. In addition to the separation procedure, we also implemented the rounding heuristic proposed in [87] and employed it to compute lower bounds.

Table 5.1 summarizes the preliminary experiments with some instances from the DIMACS benchmark and for random graphs with 100 vertices (last two rows). The notation $G(n, d)$ specifies random graphs with n vertices and a density of $d \in [0, 1]$, and for these instances we report the average results over five randomly-generated instances. The experiments were performed on a 32-bit personal computer, with a time limit of five minutes. The preprocessing, cut generation, and variable fixing procedures from Cplex are turned off.

Following the approach used in [88], for each graph we choose the best parameters and report the obtained results. The first four columns contain the instance name, the number of vertices, the graph density, and its stability number. The following three columns contain data for the root note in the enumeration tree: the column "LB" contains the lower bound found by the rounding heuristic, the column "UB" contains the upper bound after the last successful execution of the cut-generating procedure, and the column "Time" reports the total time spent at the root node, in seconds. The cells marked with $***$ specify that the computation has been aborted at the time limit. The columns UB[88] and UB[87] contain the best upper bound attained in [88] and [87], respectively.

Table 5.2 contains the number of generated clique cuts, violated rank inequalities, and violated generalized rank inequalities, respectively. As this Table shows, the procedure is able to generate a large number of cuts, and provides upper bounds that are competitive with those generated in [87] and [88] for a representative sample of benchmark graphs. Similarly to existing procedures, out cut-generating algorithm finds a large number of violated clique inequalities, and is also able to find many violated rank inequalities. The number of generalized rank inequalities generated by the procedure is smaller, but nevertheless provides an interesting set of additional and non-trivial valid inequalities.

5.7 Resumen del capítulo

Dado un grafo G, en el problema del *conjunto independiente máximo* (CIM) buscamos un conjunto independiente S en G de cardinalidad máxima. CIM es NP-hard, y fue analizado en la literatura utilizando diversas técnicas. Fueron diseñados varios métodos exactos para resolverlo, ver [14] para un compendio de los mismos.

A pesar de que los métodos combinatorios para el CIM (como los de [91, 94]) son más eficientes que los algoritmos branch and cut, es de gran interés continuar la búsqueda de métodos poliedrales eficientes para este problema. Esto se debe a que (a) CIM aparece frecuentemente como subestructura en varios problemas de optimización combinatoria, (b) en varias situaciones CIM es resuelto como subrutina para generar desigualdades válidas para modelos de programación entera (ver por ejemplo, [3]), y (c) aplicaciones reales pueden necesitar versiones específicas de CIM con restricciones adicionales y en este contexto la programación entera a menudo resulta efectiva.

Mannino y Sassano [82] introdujeron en 1996 la idea de proyecciones de aristas como una especialización de la operación de proyección de cliques de Lovász y Plummer [79]. Varias propiedades de proyección de aristas fueron estudiadas en [82] y, basado en estas propiedades, fue desarrollado un método para computar una cota superior para CIM. Esta cota es utilizada luego en un esquema de branch and bound. Rossi y Smriglio incorporan estas ideas en un entorno de programación lineal en [88], donde propusieron un procedimiento de separación basado en proyecciones de aristas. Este procedimiento elimina y proyecta en forma iterativa aristas con ciertas propiedades, y encuentra en forma heurística desigualdades de rango violadas (i.e., desigualdades de la forma $\sum_{v\in A} x_v \leq \alpha(G[A])$, donde $A \subseteq V$ and $G[A]$ es el subgrafo de G inducido por A). Pardalos et al. [87] extendieron la teoría de proyección de aristas estudiando las propiedades de facetitud obtenidas con este procedimiento. Los autores dan un algoritmo branch and cut que utiliza proyecciones de aristas como una herramienta de separación, así como varias familias de desigualdades conocidas, como las desigualdades de ciclo impar (con un algoritmo polinomial exacto de separación), las desigualdades clique (con heurísticas) y cortes mod-$\{2, 3, 5, 7\}$.

Las proyecciones de aristas son un caso especial de las proyecciones de cliques de Lovász y Plummer [79]. Rossi y Smriglio proponen en [88] emplear una secuencia de proyecciones de aristas para reducir el grafo original G y hacerlo más denso al mismo tiempo, permitiendo de esta manera una identificación más rápida de desigualdades clique en el grafo reducido G'. Un paso clave para lograr esto es establecer de qué manera $\alpha(G)$ es afectada por estas proyecciones, o, en otras palabras, exactamente cómo $\alpha(G)$ se relaciona con $\alpha(G')$. Nuestro objetivo es generalizar el procedimiento de Rossi y Smriglio proyectando cliques en lugar de aristas, y para ello necesitamos mostrar cómo se modifica $\alpha(G)$ luego de esta operación. Nuestro método permite además establecer una relación más general entre G y G'.

En este capítulo proponemos el uso de proyecciones de cliques como método general para la generación de planos de corte para el CIM, así como un nuevo procedimiento de lifting de cliques que conduce a desigualdades más generales que las obtenidas con el método de proyección de aristas. El método propuesto es capaz de generar tanto desigualdades de rango como desigualdades de rango generalizadas, recurriendo al procedimiento general de lifting introducido en [100]. Este enfoque se aparta del paradigma usual basado en templates para generar cortes, y procura unificar y generalizar el procedimiento de separación para los cortes conocidos. En este sentido, nuestro principal objetivo es aportar una mejor comprensión del polítopo del CIM, que puede contribuir también a la solución de otros problemas en optimización combinatoria.

Sea $n = |V|$ y $\mathcal{S}(G) \subseteq \{0,1\}^n$ el conjunto todos los vectores característicos de conjuntos independientes de G. Escribiremos simplemente $\mathcal{S}$ cuando G se sobreentienda por contexto. Para $W \subseteq V$, $\mathcal{S}(G[W])$ denota los vectores característicos de conjuntos independientes de G que solamente involucren vértices de W. El polítopo de conjuntos independientes de G es denotado por

$$STAB(G) = \mathrm{conv}\{x \mid x \in \mathcal{S}(G)\}.$$

Definición (Proyección de cliques [79]): Sea $W \subseteq V$, $|W| \geq 2$, una clique en G. La *proyección de cliques* de W da como resultado el grafo $G \mid W = (V, E \mid W)$, donde $E \mid W = E \cup \{xy \notin E \mid W \subseteq N(x) \cup N(y)\}$.

Lema (Lema de lifting [100]): Sea $U \subseteq V$, $\beta \in \mathbb{R}$ y $\pi_v \in \mathbb{R}$, para todo $v \in U$, tal que $\sum_{v \in U} \pi_v x_v \leq \beta$ es una desigualdad válida para $STAB(G)$. Si $c^\top x - d \leq 0$, $c, x \in \mathbb{R}^n$ y $d \in \mathbb{R}$, es una desigualdad válida para $F_U(\beta, \pi_{v \in U})$, entonces

$$L_{x,\lambda}(U) = (c^\top x - d) - \lambda \left(\sum_{v \in U} \pi_v x_v - \beta \right) \leq 0, \tag{5.8}$$

con

$$\lambda \leq \min \left\{ \frac{c^\top x - d}{\sum_{v \in U} \pi_v x_v - \beta} \mid x \in \mathcal{S}, \sum_{v \in U} \pi_v x_v < \beta \right\}, \tag{5.9}$$

es una desigualdad válida para $STAB(G)$.

Se define $\mathbf{1}_{v \in W}$ como el vector binario y de tamaño n tal que $y[v] = 1$ si y sólo si $v \in W$. Un caso especial del Lema anterior ocurre cuando U es una clique en G tal que $U = W \cup N_W$. En ese caso, si $c^\top x - d \leq 0$ es una desigualdad válida para $F_{W \cup N_W}(1, \mathbf{1}_{v \in W \cup N_W})$, entonces la aplicación de la operación de lifting (5.8) para cualquier λ válido es denominada *lifting de clique* de $c^\top x - d \leq 0$ con respecto a W. El siguiente Lema establece una condición suficiente para que una operación de clique lifting resulte en una desigualdad válida para $STAB(G)$.

Lema: Sea $W \subseteq V$, $|W| \geq 2$ una clique en G y $c^\top x - d \leq 0$, $c, x \in \mathbb{R}^n$ y $d \in \mathbb{R}$ una desigualdad válida para $STAB(G \mid W)$ con soporte H. Sea además $c_{min} = \min\{c_v \mid v \in H \setminus (W \cup N_W), c_v > 0\}$. Si W contiene un vértice w tal que $N(w) \setminus W$ es una clique en G, entonces

$$\lambda = d - c_{min}\alpha(G[H \setminus (W \cup N_W)]) - \max_{x \in \mathcal{S}(G[H \setminus (W \cup N_W)])} \sum_{v \in H \setminus (W \cup N_W)} \max\{0, c_v - c_{min}\} x_v \tag{5.10}$$

es válida para el lifting de clique de $c^\top x - d \leq 0$ con respecto a W.

Presentamos finalmente el algoritmo de generación de planos de corte, el principal resultado de este capítulo.

Algorithm 3 Procedimiento de generación de planos de corte

Encontrar un clique inicial $W^{(0)}$ de G;
$k := 0$;
while $x(W^{(k)}) \leq 1$ **do**
  Eliminar aristas tales que $N(w) \backslash W^{(k)}$ es una clique, para algún $w \in W^{(k)}$;
  Proyectar la clique $W^{(k)}$, obteniendo el grafo $G^{(k+1)}$;
  Encontrar una clique $W^{(k+1)}$;
  $k := k + 1$;
end while
Sea $\pi^{(k)}$ el vector característico de $W^{(k)}$ (tal que la desigualdad $\pi^{(k)} x \leq \gamma^{(k)} := 1$ sea violada);
for $i \leftarrow k, \ldots, 1$ **do**
  Aplicar el Lema anterior a $\pi^{(k)} x \leq \gamma^{(k)}$ y $W^{(k)}$ en el grafo $G^{(k-1)}$, obteniendo una nueva desigualdad $\pi^{(k-1)} x \leq \gamma^{(k-1)}$ válida para $G^{(k-1)}$;
end for
return $\pi^{(0)} x \leq \gamma^{(0)}$, si es violada;

CHAPTER 6

Conclusions

We presented in Chapter 2 a simple linear time algorithm to compute the k, i-chromatic number and an optimum k, i-coloring of cycles and cacti. We have also adapted the algorithm in order to obtain an optimum $(k : i)$-coloring of cycles and cacti. As a future line of work, we will explore the possible extension of these ideas to graph subdivisions and graphs with large girth. We also studied in this chapter Vizing's theorem $\chi(G \square H) = max\{\chi(G), \chi(H)\}$ applied to k, i-colorings for some special graph products. It would be interesting to know if this result does hold in general for two arbitrary graphs G and H.

In Chapter 3, we analyzed the b-coloring problem on P_4-tidy graphs. We showed that these graphs are b-continuous and b-monotonic. We also described a polynomial time algorithm to compute their b-chromatic number. The algorithm runs in $O(n^3)$ time; further research on this might lead to improvements on the running time. Also, there are already interesting generalizations of P_4-tidy graphs defined in the literature, such as graphs with a limited number of partners [89]. A next step for the work in this chapter could be to analyze the generalization of the coloring algorithm and the parameters of b-continuity and b-monotonicity for this graph class.

Chapter 4 was focused on the b-edge coloring problem for the direct product of graphs. We gave bounds for the b-chromatic index of the direct product of a graph G and a regular graph H admitting a partition of its edges into perfect matchings. Next, we presented exact results of χ'_b for the direct product of some connected regular graphs. Finally, we developed a simple integer linear programming model for the problem. With this method and all previous results we were able to produce exact values of χ'_b for the direct product of paths and for the direct product of cycles. We see potential for improvement for the integer linear programming model given, in the following three directions: first, the feasibility of a cutting-plane procedure could be analyzed for increasing performance of the model. Second, an attempt could be made to convert further properties of the problem into valid inequalities. Third, alternative models could be explored, following the example of the several integer programming formulations for classic graph coloring existing in the literature.

We presented in Chapter 5 a general cut-generating procedure for the standard formulation of the maximum stable set polytope, which is able to generate both violated rank and generalized rank inequalities. The main objective of this algorithm is to generalize existing procedures based on edge projection, and employs a lifting procedure in order to construct general valid inequalities from an initial clique inequality by undoing the operation of clique projection in the original graph. The computational experiments presented are of a preliminary nature, and show that the proposed procedure is effective at generating general cuts, and may be competitive in a general setting. As a future work, we intend to perform extensive computational experiments with the proposed cut generating procedure in a full branch and cut algorithm. Additionally, this work could be connected to the graph coloring problem by using the cutting plane procedure proposed on the standard formulation for classic coloring, instead of the maximum stable set problem. All the inequalities generated by the procedure would be valid for the coloring problem, since each color class constitutes a stable set. It would be interesting to see if this approach is competitive.

6.1 Resumen del capítulo

Presentamos en el Capítulo 2 un algoritmo sencillo para computar el número k, i-cromático y un k, i-coloreo óptimo de ciclos y cactus. Adaptamos también el algoritmo para obtener un $(k : i)$-coloreo óptimo de ciclos y cactus. Como una posible línea de trabajo futuro, exploraremos la posible extensión de estas ideas a subdivisiones de grafos y grafos de cintura ancha. Estudiamos además en este capítulo el teorema de Vizing $\chi(G \Box H) = max\{\chi(G), \chi(H)\}$ aplicado a k, i-coloreos para algunos productos de grafos. Sería interesante saber si este resultado vale en general para dos grafos arbitrarios G y H.

En el Capítulo 3, analizamos el problema de b-coloreo en grafos P_4-tidy. Mostramos que estos grafos son b-continuos y b-monótonos. Describimos también un algoritmo polinomial para computar el número b-cromático. El algoritmo tiene un orden de complejidad de $O(n^3)$; una continuación del estudio del mismo podría conducir a mejoras en el tiempo de ejecución. Además, existen actualmente en la literatura generalizaciones interesantes de los grafos P_4-tidy, como los grafos con un número limitado de partners [89]. Un paso siguiente para el trabajo en este capítulo podría ser intentar generalizar el algoritmo de coloreo y estudiar los parámetros de b-continuidad y b-monotonía para esta clase de grafos.

El Capítulo 4 se focalizó en el problema de b-coloreo de aristas para el producto directo de grafos. Dimos cotas para el índice b-cromático del producto directo de un grafo G y un grafo regular H que admite una partición de sus aristas en matchings perfectos. Luego, presentamos resultados exactos para χ'_b para el producto directo de algunos grafos regulares conexos. Finalmente, desarrollamos un sencillo modelo de programación lineal entera para el problema. Con este modelo y los resultados anteriores nos fue posible calcular en forma exacta χ'_b para el producto directo de caminos y el producto directo de ciclos. Vemos potencial de mejora en el modelo, en los siguientes sentidos: primero, el análisis de un procedimiento de planos de corte para mejorar la eficiencia del modelo. Segundo, podrían ser explorados modelos alternativos, siguiendo el ejemplo de las diversas formulaciones en programación lineal entera que existen en la literatura para el problema del coloreo clásico de grafos.

Presentamos en el Capítulo 5 un procedimiento general de generación de planos de corte para el polítopo del problema de conjunto independiente máximo, que es capaz de generar desigualdades violadas, de rango y de rango generalizadas. El objetivo principal de este algoritmo es generalizar los métodos existentes basados en proyección de aristas. El algoritmo emplea un procedimiento de lifting para construir desigualdades válidas generales a partir de una desigualdad clique inicial, deshaciendo la operación de proyección de cliques del grafo original. Los experimentos computacionales presentados son de naturaleza preliminar, y muestran que el procedimiento propuesto es efectivo generando cortes, y puede ser competitivo. En un trabajo futuro, realizaremos experimentos computacionales extensivos, integrando el procedimiento de corte en un esquema branch and cut completo. Adicionalmente, este trabajo puede relacionarse con el problema de coloreo de grafos utilizando el procedimiento de cortes propuesto para la formulación estándar de programación lineal entera para el coloreo clásico, en lugar del modelo para el problema del conjunto independiente máximo. Todas las desigualdades generadas por el procedimiento continuarían siendo válidas para el problema de coloreo, ya que cada clase color constituye un conjunto independiente. Sería de interés saber si este enfoque puede resultar competitivo.

Bibliography

[1] E. Agrell, A. Vardy, and K. Zeger, Upper bounds for constant-weight codes, *IEEE Transactions on Information Theory* **46**(7) (2000), 2373–2395.

[2] M. Alkhateeb and A. Kohl, Upper bounds on the b-chromatic number and results for restricted graph classes, *Discussiones Mathematicae: Graph Theory* **31**(4) (2011), 709–735.

[3] A. Atamtürk, G. L. Nemhauser, and M. W. P. Savelsbergh, Conflict graphs in solving integer programming problems, *European Journal of Operational Research* **121**(1) (2000), 40–55.

[4] L. Babel and S. Olariu, On the structure of graphs with few P_4s, *Discrete Applied Mathematics* **84** (1998), 1–13.

[5] L. Babel and S. Olariu, On the p-connectedness of graphs - A survey, *Discrete Applied Mathematics* **95**(1–3) (1999), 11–33.

[6] R. Balakrishnan and S. Francis Raj, Bounds for the b-chromatic number of $G-v$, *Discrete Applied Mathematics* **161**(9) (2013), 1173–1179.

[7] R. Balakrishnan, S. Francis Raj, and T. Kavaskar, *Bounds for the b-chromatic number of the Mycielskian of some families of graphs*, To appear in Ars Combinatoria.

[8] E. Balas and M. Padberg, Set partitioning: a survey, *SIAM Review* **18** (1976), 710–760.

[9] D. Barth, J. Cohen, and T. Faik, On the b-continuity property of graphs, *Discrete Applied Mathematics* **155**(13) (2007), 1761–1768.

[10] C. Berge, *Graphs and Hypergraphs*, North–Holland, Amsterdam, 1976.

[11] C. Betancur Velasquez, F. Bonomo, and I. Koch, On the b-coloring of P_4-tidy graphs, *Discrete Applied Mathematics* **159**(1) (2011), 60–68.

[12] M. Blidia, F. Maffray, and Z. Zemir, On b-colorings in regular graphs, *Discrete Applied Mathematics* **157**(8) (2009), 1787–1793.

[13] B. Bollobás and A. Thomason, Set colourings of graphs, *Discrete Mathematics* **25**(1) (1979), 21–26.

[14] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, The maximum clique problem, *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1999, pp. 1–74.

[15] F. Bonomo, G. Durán, I. Koch, and M. Valencia-Pabon, *On the* (k, i)*-coloring of cacti and complete graphs*, Submitted to Ars Combinatoria.

[16] F. Bonomo, G. Durán, F. Maffray, J. Marenco, and M. Valencia-Pabon, On the b-coloring of cographs and P_4-sparse graphs, *Graphs and Combinatorics* **25**(2) (2009), 153–167.

[17] F. Bonomo, O. Schaudt, M. Stein, and M. Valencia-Pabon, b-coloring is NP-hard on co-bipartite graphs and polytime solvable on tree-cographs, *Proceedings of the International Symposium on Combinatorial Optimization 2014* (L. Gouveia and A. Ridha Mahjoub, eds.), Lecture Notes in Computer Science, 2014.

[18] R. C. Brigham and R. D. Dutton, Generalized k-tuple colorings of cycles and other graphs, *Journal of Combinatorial Theory. Series B* **32** (1982), 90–94.

[19] E. Burke, D. De Werra, and J. Kingston, Applications to timetabling, *Handbook of Graph Theory* (J. L. Gross and J. Yellen, eds.), CRC Press, 2004, pp. 445–475.

[20] S. Cabello and M. Jakovac, On the b-chromatic number of regular graphs, *Discrete Applied Mathematics* **159** (2011), 1303–1310.

[21] M. Campêlo, V. Campos, and R. Corrêa, On the asymmetric representatives formulation for the vertex coloring problem, *Discrete Applied Mathematics* **156**(7) (2008), 1097–1111.

[22] V. Campos, V. Farias, and A. Silva, b-coloring graphs with large girth, *Journal of the Brazilian Computer Society* **18**(4) (2012), 375–378.

[23] V. Campos, C. Linhares Sales, F. Maffray, and A. Silva, b-chromatic number of cacti, *Electronic Notes in Discrete Mathematics* **35** (2009), 281–286.

[24] V. Campos, C. Linhares Sales, A. Maia, and R. Sampaio, Maximization Coloring Problems on graphs with few P_4s, *Discrete Applied Mathematics* **164**(2) (2014), 539–546.

[25] F. Chaouche and A. Berrachedi, Some bounds for the b-chromatic number of generalized Hamming graphs, *Far East Journal of Applied Mathematics* **26** (2007), 375–391.

[26] G. Chartrand and P. Zhang, *Chromatic graph theory*, 1st ed., Chapman & Hall/CRC, 2008.

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

[28] D. Corneil, Y. Perl, and L. Stewart, Cographs: recognition, applications and algorithms, *Congressus Numerantium* **43** (1984), 249–258.

[29] R. Corrêa, I. Koch, and J. Marenco, A general procedure for generating cuts for the maximum stable set problem, *Submitted to VIII ALIO/EURO Workshop on Applied Combinatorial Optimization, Montevideo, Uruguay*, 2014.

[30] S. Corteel, M. Valencia-Pabon, and J. Vera, On approximating the b-chromatic number, *Discrete Applied Mathematics* **146**(1) (2005), 618–622.

[31] T. Denley, The odd girth of the generalized Kneser graph, *European Journal of Combinatorics* **18**(6) (1997), 607–611.

[32] B. Effantin, The b-chromatic number of power graphs of complete caterpillars, *Journal of Discrete Mathematical Sciences & Cryptography* **8** (2005), 483–502.

[33] B. Effantin and H. Kheddouci, The b-chromatic number of some power graphs, *Discrete Mathematics & Theoretical Computer Science* **6**(1) (2003), 45–54.

[34] B. Effantin and H. Kheddouci, Exact values for the b-chromatic number of a power complete k-ary tree, *Journal of Discrete Mathematical Sciences & Cryptography* **8** (2005), 117–129.

[35] A. El-Sahili and M. Kouider, About b-colourings of regular graphs, Tech. Report 1432, Unité Mixte de Recherche 8623, CNRS-Université Paris Sud LRI, 2006.

[36] S. Fiorini and R. J. Wilson, *Edge colourings of graphs*, Research Notes in Mathematics, Pitman, 1977.

[37] I. Fister, I. Peterin, M. Mernik, and M. Črepišek, *Hybrid evolutionary algorithm for the b-chromatic number*, Manuscript 2013.

[38] V. Giakoumakis, F. Roussel, and H. Thuillier, On P_4-tidy graphs, *Discrete Mathematics & Theoretical Computer Science* **1** (1997), 17–41.

[39] C. D. Godsil and G. Royle, *Algebraic graph theory*, Graduate Texts in Mathematics, Springer, 2001.

[40] R. L. Graham and N. J. A. Sloane, Lower bounds for constant weight codes, *IEEE Transactions on Information Theory* **26** (1980), 37–43.

[41] R. Hammack, A quasi cancellation property for the direct product, *Discrete Mathematics* **310** (2010), 1691–1696.

[42] R. Hammack, W. Imrich, and S. Klavžar, *Handbook of product graphs*, CRC Press, 2011.

[43] H. Hanani, On quadruple systems, *Canadian Journal of Mathematics* **12** (1960), 145–157.

[44] H. Hanani, The existence and construction of balanced incomplete block designs, *Annals of Mathematical Statistics* **32** (1961), 361–386.

[45] H. Hanani, On some tactical configurations, *Canadian Journal of Mathematics* **15** (1963), 702–722.

[46] H. Hanani, A balanced incomplete block design, *Annals of Mathematical Statistics* **36** (1965), 711.

[47] F. Harary, S. Hedetniemi, and G. Prins, An interpolation theorem for graphical homomorphisms, *Portugaliae mathematica* **26** (1967), 453–462 (in English).

[48] F. Havet, C. Linhares-Sales, and L. Sampaio, b-coloring of tight graphs, *Discrete Applied Mathematics* **160**(18) (2012), 2709–2715.

[49] S. T. Hedetniemi, Homomorphisms of graphs and automata, Tech. Report 03105-44-T, University Michigan, Ann Arbor, MI., 1966.

[50] A. Hilton, R. Rado, and S. Scott, A (< 5)-colour theorem for planar graphs, *Bulletin of the London Mathematical Society* **5** (1973), 302–306.

[51] C. T. Hoàng, *Perfect graphs*, Ph.D. Thesis, School of Computer Science, McGill University, Montreal, 1985.

[52] C. T. Hoàng and M. Kouider, On the b-dominating coloring of graphs, *Discrete Applied Mathematics* **152** (2005), 176–186.

[53] C. T. Hoàng, C. Linhares Sales, and F. Maffray, On minimally b-imperfect graphs, *Discrete Applied Mathematics* **157**(17) (2009), 3519–3530.

[54] C. T. Hoàng, F. Maffray, and M. Mechebbek, A characterization of b-perfect graphs, *Journal of Graph Theory* **71**(1) (2012), 95–122.

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

[56] R. W. Irving and D. F. Manlove, The b-chromatic number of a graph, *Discrete Applied Mathematics* **91** (1999), 127–141.

[57] M. Jakovac and S. Klavžar, The b-chromatic number of cubic graphs, *Graphs and Combinatorics* **26**(1) (2010), 107–118.

[58] M. Jakovac and I. Peterin, *b-chromatic index of a graph*, To appear in Bulletin of the Malaysian Mathematical Sciences Society.

[59] M. Jakovac and I. Peterin, On the b-chromatic number of some products, *Studia Scientiarum Mathematicarum Hungarica* **49** (2012), 156–169.

[60] B. Jamison and S. Olariu, A new class of brittle graphs, *Studies in Applied Mathematics* **81** (1989), 89–92.

[61] B. Jamison and S. Olariu, P_4-reducible graphs – a class of uniquely tree representable graphs, *Studies in Applied Mathematics* **81** (1989), 79–87.

[62] B. Jamison and S. Olariu, On a unique tree representation for P_4-extendible graphs, *Discrete Applied Mathematics* **34** (1991), 151–164.

[63] B. Jamison and S. Olariu, p-components and the homogeneous decomposition of graphs, *SIAM Journal on Discrete Mathematics* **8** (1995), 448–463.

[64] R. Javadi and B. Omoomi, On b-coloring of the Kneser graphs, *Discrete Mathematics* (2009), 4399–4408.

[65] R. Javadi and B. Omoomi, On b-coloring of cartesian product of graphs., *Ars Combinatoria* **107** (2012), 521–536.

[66] T. Jensen and B. Toft, *Graph coloring problems*, John Wiley & Sons, New York, 1995.

[67] S. M. Johnson, A new upper bound for error-correcting codes, *IEEE Transactions on Information Theory* **8** (1962), 203–207.

[68] J. Kára, J. Kratochvíl, and M. Voigt, b-continuity, Tech. Report M 14/04, Technical University Ilmenau, Faculty of Mathematics and Natural Sciences, 2004.

[69] R. Karp, Reducibility among combinatorial problems, In: *Complexity of Computer Computations* (R. Miller and J. Thatcher, eds.), Plenum Press, New York, 1972, pp. 85–103.

[70] L. G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* **20** (1979), 191–194.

[71] A. Khelladi and C. Payan, Generalized n-tuple colorings of a graph: A counterexample to a conjecture of Brigham and Dutton, *Journal of Combinatorial Theory. Series B* **37** (1984), 283–289.

[72] S. R. Kim, Centers of a tensor composite graph, *Congressus Numerantium* **81** (1991), 193–203.

[73] S. Klein and M. Kouider, On b-perfect graphs, *Annals of the XII Latin-Ibero-American Congress on Operations Research*, Havanna, Cuba, October 2004.

[74] I. Koch and I. Peterin, *The b-chromatic index of direct product of graphs*, Submitted to Discrete Applied Mathematics, 2014.

[75] M. Kouider and M. Maheo, Some bounds for the b-chromatic number of a graph, *Discrete Mathematics* **256** (2002), 267–277.

[76] M. Kouider and M. Zaker, Bounds for the b-chromatic number of some families of graphs, *Discrete Mathematics* **306** (2006), 617–623.

[77] J. Kratochvíl, Zs. Tuza, and M. Voigt, On the b-chromatic number of a graph, *Lecture Notes in Computer Science* **2573** (2002), 310–320.

[78] C. V. G. C. Lima, N. A. Martins, L. Sampaio, M. C. Santos, and A. Silva, b-chromatic index of graphs, *Electronic Notes in Discrete Mathematics* **44** (2013), 9–14.

[79] L. Lovász and M. D. Plummer, *Matching theory*, Akadémiai Kiadó, Budapest, 1986.

[80] F. Maffray and M. Mechebbek, On b-perfect chordal graphs, *Graphs and Combinatorics* **25**(3) (2009), 365–375.

[81] F. Maffray and A. Silva, b-coloring outerplanar graphs with large girth, *Discrete Mathematics* **312**(10) (2012), 1796–1803.

[82] C. Mannino and A. Sassano, Edge projection and the maximum cardinality stable set problem, In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 26, American Mathematical Society, 1996, pp. 249–261.

[83] I. Méndez-Díaz and P. Zabala, A generalization of the graph coloring problem, *Investigación Operativa* **8** (1999), 167–184.

[84] I. Méndez-Díaz and P. Zabala, Solving a multicoloring problem with overlaps using integer programming, *Discrete Applied Mathematics* **158** (2010), 349–354.

[85] R. Murphey, P. Pardalos, and M. Resende, Frequency assignment problems, In: *Handbook of Combinatorial Optimization* (D. Du and P. Pardalos, eds.), Springer US, 1999, pp. 295–377.

[86] P. Östergaard, A new algorithm for the maximum weight clique problem, *Nordic Journal of Computing* **8**(4) (2001), 424–436.

[87] S. Rebennack, M. Oswald, D. O. Theis, H. Seitz, G. Reinelt, and P. M. Pardalos, A branch and cut solver for the maximum stable set problem, *Journal of Combinatorial Optimization* **21** (2011), 434–457.

[88] F. Rossi and S. Smriglio, A branch-and-cut algorithm for the maximum cardinality stable set problem, *Operations Research Letters* **28** (2001), 63–74.

[89] F. Roussel, I. Rusu, and H. Thuillier, On graphs with limited number of P_4-partners., *International Journal of Foundations of Computer Science* **10**(1) (1999), 103–122.

[90] A. Schrijver, *Combinatorial optimization. Polyhedra and efficiency (3 volumes)*, Algorithms and Combinatorics, vol. 24, Springer–Verlag, Berlin, 2003.

[91] P. S. Segundo, D. Rodríguez-Losada, and A. Jiménez, An exact bit-parallel algorithm for the maximum clique problem, *Journal of Computers and Operations Research* **38** (2011), 571–581.

[92] S. Stahl, n-Tuple colorings and associated graphs, *Journal of Combinatorial Theory. Series B* **20** (1976), 185–203.

[93] K. Thilagavathi, M. D. Vijayalakshmi, and M. N. Roopesh, b-colouring of central graphs, *International Journal of Computer Applications* **3**(11) (2010), 27–29.

[94] E. Tomita and T. Kameda, An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments, *Journal of Global Optimization* **37**(1) (2007), 95–111.

[95] V. Vernold, The b-chromatic number of star graph families, *Le Matematiche* **65**(1) (2010), 119–125.

[96] D. Vijayalakshmi, K. Thilagavathi, and N. Roopesh, b-chromatic number of $M[C_n], M[P_n], M[F_{1,n}]$ and $M[W_n]$, *Open Journal of Discrete Mathematics* **1**(2) (2011), 85–88.

[97] V. G. Vizing, The cartesian product of graphs, *Vychislitelnye Sistemy* **9** (1963), 30–43.

[98] P. M. Weichsel, The Kronecker product of graphs, *Proceedings of the American Mathematical Society*, vol. 13, 1962, pp. 47–52.

[99] R. Wilson, Graph theory, In: *History of Topology* (I.M. James, ed.), North-Holland, Amsterdam, 1999, pp. 503–529.

[100] A. S. Xavier and M. B. Campêlo, A new facet generating procedure for the stable set polytope, *Electronic Notes in Discrete Mathematics* **37** (2011), 183–188.

Index

C_n, *see* cycle
$E(\cdot)$, 8, 12
$G\Box H$, *see* cartesian product
$G \times H$, *see* direct product
K_n, *see* complete graph
$L(\cdot)$, *see* line graph
$N(\cdot)$, *see* neighborhood
P_4-sparse graph, 37
P_4-tidy graph, 37
P_n, *see* path
S_n, *see* stable set
$V(\cdot)$, 8
$\Delta(\cdot)$, *see* maximum degree
$\alpha(G)$, *see* stability number
$\chi'(\cdot)$, *see* chromatic index
$\chi(\cdot)$, *see* chromatic number
$\chi'_b(\cdot)$, *see* b-chromatic index
$\chi_b(\cdot)$, *see* b-chromatic number
$\chi^i_k(\cdot)$, *see* k,i-chromatic number
$\chi^{(i)}_k(\cdot)$, 16
$\delta(\cdot)$, *see* minimum degree
$\overline{G}$, *see* complement graph
$d_G(u,v)$, *see* distance
$diam(G)$, *see* diameter

b-continuous, 36
b-monotonic, 36
b-perfect, 37
bipartite graph, 8
block, 9
branch and bound, 11
branch and cut, 11

cactus, 26
cartesian product, 26
chromatic index, 9
    b-chromatic index, 53
chromatic number, 9
    k,i-chromatic number, 16
    b-chromatic number, 35
class 1 graphs, class 2 graphs, 57
clique, 8
clique projection, 70
cograph, 37
color class, 9
coloring
    $(k:i)$-coloring, 16
    k,i-coloring, 16
    b-coloring, 35
    b-edge coloring, 53
    edge coloring, 9
    tuple coloring, 16
    vertex coloring, 9
complement graph, 8
complete graph, 8
connected
    component, 9
    graph, 9
constraint, 10
convex hull, 10
convex set, 10
cut-vertex, 9
cutting plane algorithm, 11

degree, 8
diameter, 9
direct product, 53
distance, 9
dominant vertex, edge, 9

edge, 8
edge projection, 71
empty graph, 8
end-block, 9
endpoint, 8

face, 10

facet, 10
  defining inequality, 10
false edges, 71
fiber, 26
forest, 9

generalized Kneser graph, 19
girth, 9
graph, 8

homomorphism, 19

independent set, *see* stable set
induced subgraph, 8

line graph, 8
linear programming problem, 10
  integer, 10

matching, 8
maximum degree, 8
maximum stable set problem, 68
minimum degree, 8
multicycle, 20
multigraph, 8
  uniform, 20

neighborhood, 8
  closed, 8

objective function, 10
one-factor, *see* perfect matching

partner, 37
path, 9
perfect matching, 54
polyhedron, 10
polytope, 10
projection of an edge, 53

regular graph, 8

simplicial, 38
stability number, 8
stable set, 8
subgraph, 8

tree, 9
twin
  false, 38
  true, 38

valid inequality, 10
vertex, 8

walk, 9