

VARIATIONS OF COLORING PROBLEMS RELATED TO SCHEDULING AND DISCRETE TOMOGRAPHY

THÈSE N° 3968 (2007)

PRÉSENTÉE LE 14 DÉCEMBRE 2007

À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE DE RECHERCHE OPÉRATIONNELLE SE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Bernard RIES

ingénieur mathématicien diplômé EPF
et de nationalité luxembourgeoise

acceptée sur proposition du jury:

Prof. T. Mountford, président du jury
Prof. D. de Werra, directeur de thèse
Prof. M. Bierlaire, rapporteur
Dr F. Maffray, rapporteur
Prof. C. Picouleau, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2007

Acknowledgements

I would like to thank the following people for their support during my thesis :

First of all, my supervisor Professor Dominique de Werra for having proposed me to do a Ph.D. thesis in his research group ROSE. It has been a real pleasure to work with him and I'm grateful for all the useful advices he gave me and for the time he spent together with me on this thesis.

Dr Cédric Bentz, Professor Marie-Christine Costa and Professor Christophe Picouleau for having passed so much time with me in the beautiful world of discrete tomography.

Nicolas Zufferey for having been so kind to revise this thesis with great care. Michel Bierlaire and Frédéric Maffray for having accepted to be members of the jury. Furthermore I would like to thank Michel for always being available and for all the useful advices he gave me.

The former and present members of ROSE, for the nice atmosphere they have created and for making this group something special ; particularly Tinaz for all her good advices and her patience ; Benji for his organizing talents and for explaining me how to get along with a computer ; Sivan for being available all time ; Jocelyne for arranging all our trips and for her positive energy ; Simona, Daniela, Ivo, David, Nicolas and Sacha for their friendship and the nice time that we passed together.

My family, especially my parents for having supported me during my whole studies and my thesis and for always believing in me.

My friends in Lausanne and in Luxembourg for encouraging me all the years and for the unforgettable moments spent together.

Last but not least I would like to thank Isabelle for her love, her support and her unbreakable optimism every day !

Abstract

The graph coloring problem is one of the most famous problems in graph theory and has a large range of applications. It consists in coloring the vertices of an undirected graph with a given number of colors such that two adjacent vertices get different colors. This thesis deals with some variations of this basic coloring problem which are related to scheduling and discrete tomography. These problems may also be considered as partitioning problems. In Chapter 1 basic definitions of computational complexity and graph theory are presented. An introduction to graph coloring and discrete tomography is given.

In the next chapter we discuss two coloring problems in mixed graphs (i.e., graphs having edges and arcs) arising from scheduling. In the first one (strong mixed graph coloring problem) we have to cope with disjunctive constraints (some pairs of jobs cannot be processed simultaneously) as well as with precedence constraints (some pairs of jobs must be executed in a given order). It is known that this problem is \mathcal{NP} -complete in mixed bipartite graphs. In this thesis we strengthen this result by proving that for $k = 3$ colors the strong mixed graph coloring problem is \mathcal{NP} -complete even if the mixed graph is planar bipartite with maximum degree 4 and each vertex incident to at least one arc has maximum degree 2 or if the mixed graph is bipartite and has maximum degree 3. Furthermore we show that the problem is polynomially solvable in partial p -trees, for fixed p , as well as in general graphs with $k = 2$ colors. We also give upper bounds on the strong mixed chromatic number or even its exact value for some classes of graphs. In the second problem (weak mixed graph coloring problem), we allow jobs linked by precedence constraints to be executed at the same time. We show that for $k = 3$ colors this problem is \mathcal{NP} -complete in mixed planar bipartite graphs of maximum degree 4 as well as in mixed bipartite graphs of maximum degree 3. Again, for partial p -trees, p fixed, and for general graphs with $k = 2$ colors, we prove that the weak mixed graph coloring problem is polynomially solvable.

We consider in Chapter 3 the problem of characterizing in an undirected graph $G = (V, E)$ a minimum set R of edges for which maximum matchings M can be found with specific values of $p = |M \cap R|$. We obtain partial results for some classes of graphs and show in particular that for odd cacti with triangles only and for forests one can determine in polynomial time whether there exists a minimum set R for which there are maximum matchings M such that $p = |R \cap M|$, for $p = 0, 1, \dots, \nu(G)$.

The remaining chapters deal with some coloring (or partitioning) problems related to the

basic image reconstruction problem in discrete tomography.

In Chapter 4 we consider a generalization of the vertex coloring problem associated with the basic image reconstruction problem. We are given an undirected graph and a family of chains covering its vertices. For each chain the number of occurrences of each color is given. We then want to find a coloring respecting these occurrences. We are interested in both, arbitrary and proper colorings and give complexity results. In particular we show that for arbitrary colorings the problem is \mathcal{NP} -complete with two colors even if the graph is a tree of maximum degree 3. We also consider the edge coloring version of both problems. Again we present some complexity results.

We consider in Chapter 5 some generalized neighborhoods instead of chains. For each vertex x we are given the number of occurrences of each color in its open neighborhood $N_d(x)$ (resp. closed neighborhood $N_d^+(x)$), representing the set of vertices which are at distance d from x (resp. at distance at most d from x). We are interested in arbitrary colorings as well as proper colorings. We present some complexity results and we show in particular that for $d = 1$ the problems are polynomially solvable in trees using a dynamic programming approach. For the open neighborhood and $d = 2$ we obtain a polynomial time algorithm for quatrees (i.e. trees where all internal vertices have degree at least 4). We also examine the bounded version of these problems, i.e., instead of the exact number of occurrences of each color we are given upper bounds on these occurrences. In particular we show that the problem for proper colorings is \mathcal{NP} -complete in bipartite graphs of maximum degree 3 with four colors and each color appearing at most once in the neighborhood $N(x)$ of each vertex x . This result implies that the $L(1,1)$ -labelling problem is \mathcal{NP} -complete in this class of graphs for four colors.

Finally in Chapter 6 we consider the edge partitioning version of the basic image reconstruction problem, i.e., we have to partition the edge set of a complete bipartite graph into k subsets such that for each vertex there must be a given number of edges of each set of the partition incident to this vertex. For $k = 3$ the complexity status is still open. Here we present a new solvable case for $k = 3$. Then we examine some variations where the union of two subsets E^1, E^2 has to satisfy some additional constraints as for example it must form a tree or a collection of disjoint chains. In both cases we give necessary and sufficient conditions for a solution to exist. We also consider the case where we have a complete graph instead of a complete bipartite graph. We show that the edge partitioning problem in a complete graph is at least as difficult as in a complete bipartite graph. We also give necessary and sufficient conditions for a solution to exist if $E^1 \cup E^2$ form a tree or if they form a Hamiltonian cycle in the case of a complete graph. Finally we examine for both, complete and complete bipartite graphs, the case where each one of the sets E^1 and E^2 is structured (two disjoint Hamiltonian chains, two edge disjoint cycles) and present necessary and sufficient conditions.

Keywords: Graph coloring, Mixed graph, Scheduling, Discrete tomography, Edge partitioning, Complexity, Bipartite graph

Résumé

Le problème de coloration est un des plus célèbres en théorie des graphes et a de nombreuses applications. Il consiste à colorer les sommets d'un graphe non orienté de sorte que deux sommets adjacents n'aient pas la même couleur. Cette thèse traite de quelques variations du problème de coloration qui sont issues de problèmes d'ordonnancement et de la tomographie discrète. Tous ces problèmes peuvent aussi être vus comme des problèmes de partitionnement.

Dans le premier chapitre nous présentons des notions et définitions élémentaires de la théorie des graphes ainsi qu'une introduction à la coloration de graphes et à la tomographie discrète.

Dans le chapitre suivant nous considérons deux problèmes de coloration dans les graphes mixtes (i.e. des graphes contenant à la fois des arcs et des arêtes) issus de problèmes d'ordonnancement. Dans le premier ("strong mixed graph coloring problem") nous devons traiter à la fois des contraintes de disjonction (certains travaux ne peuvent pas être exercés simultanément) et des contraintes de précédence (certains travaux doivent être exécutés dans un ordre spécifique). Ce problème est \mathcal{NP} -complet dans les graphes mixtes bipartis. Dans cette thèse nous renforçons ce résultat en prouvant que pour $k = 3$ couleurs ce problème est \mathcal{NP} -complet même si le graphe mixte est biparti planaire de degré maximum 4 et les sommets incidents à au moins un arc sont de degré maximum 2 ou si le graphe mixte est biparti de degré maximum 3. De plus nous allons montrer que le problème est résoluble en temps polynomial dans les p -arbres partiels, pour p fixé, ainsi que dans les graphes mixtes avec $k = 2$ couleurs. Nous donnons également des bornes supérieures du nombre chromatique mixte voire même sa valeur exacte pour certaines classes de graphes. Dans le deuxième problème ("weak mixed graph coloring problem"), nous relâchons les contraintes de précédence en permettant aux travaux d'être exécutés en même temps. Nous montrons que pour $k = 3$ couleurs ce problème est \mathcal{NP} -complet dans les graphes mixtes bipartis planaires de degré maximum 4 ainsi que dans les graphes mixtes bipartis de degré maximum 3. À nouveau nous démontrons que dans les p -arbres partiels, p fixé, et dans les graphes avec $k = 2$ couleurs ce problème peut être résolu en temps polynomial.

Nous traitons le problème suivant dans le chapitre 3 : nous voulons caractériser dans un graphe non orienté un ensemble d'arêtes R de taille minimum tel qu'il existe des couplages maximum M avec $p = |R \cap M|$ pour des valeurs spécifiques de p . Nous allons obtenir des résultats partiels pour quelques classes de graphes et en particulier nous montrons que pour

des cactus impairs ne contenant que des triangles et pour des forêts nous pouvons déterminer en temps polynomial s'il existe un ensemble R de taille minimum pour lequel il existe des couplages maximum M avec $p = |R \cap M|$, pour $p = 0, 1, \dots, \nu(G)$.

Les trois derniers chapitres traitent de problèmes de coloration (ou de partitionnement) qui sont liés au problème basique de reconstruction d'image de la tomographie discrète.

Dans le chapitre 4, nous considérons une généralisation du problème de coloration de sommets associé au problème basique de reconstruction d'image. Étant donné un graphe non orienté et une famille de chaînes couvrant les sommets du graphe telle que pour chaque chaîne nous connaissons le nombre d'occurrences de chaque couleur, nous voulons trouver une coloration qui respecte ces occurrences. Nous nous intéressons à la fois aux colorations arbitraires et aux colorations propres et nous donnons des résultats de complexité. Nous montrons en particulier que dans le cas de la coloration arbitraire le problème est \mathcal{NP} -complet avec 2 couleurs même si le graphe est un arbre de degré maximum 3. Nous considérons également les deux problèmes dans le cas de coloration d'arêtes et présentons à nouveau des résultats de complexité.

Les chaînes sont remplacées par des voisinages généralisés dans le chapitre 5. Pour chaque sommet x , nous fixons le nombre d'occurrences de chaque couleur dans son voisinage ouvert $N_d(x)$ (resp. son voisinage fermé $N_d^+(x)$), qui contient tous les sommets qui sont à distance d de x (resp. à distance au plus d de x). Nous traitons les cas de colorations arbitraires et de colorations propres. Nous présentons des résultats de complexité et en particulier nous montrons que pour $d = 1$ les deux problèmes sont résolubles en temps polynomial en utilisant une approche de programmation dynamique. Dans le cas du voisinage ouvert et pour $d = 2$ nous obtenons un algorithme polynomial pour les "quatree" (i.e. les arbres dans lesquels tous les sommets internes sont de degré au moins 4). Nous nous intéressons également à la version bornée de ces problèmes, i.e., au lieu de connaître le nombre exact d'occurrences de chaque couleur, on se donne des bornes supérieures pour ces occurrences. Nous montrons en particulier que dans le cas de coloration propre, le problème est \mathcal{NP} -complet pour 4 couleurs dans les graphes bipartis de degré maximum 3 et chaque couleur apparaissant au plus une fois dans le voisinage $N_1(x)$ de chaque sommet x . Ce résultat implique que le problème " $L(1, 1)$ -labelling" est \mathcal{NP} -complet dans cette classe de graphes pour 4 couleurs.

Finalement dans le chapitre 6 nous considérons le problème de partitionnement d'arêtes associé au problème basique de reconstruction d'image, i.e., nous devons partitionner l'ensemble des arêtes d'un graphe biparti complet dans k sous-ensembles tels que pour chaque sommet il y ait un nombre fixé d'arêtes de chaque sous-ensemble incidentes à ce sommet. Pour $k = 3$ la complexité de ce problème n'est pas connue. Dans ce travail nous présentons un nouveau cas résoluble en temps polynomial pour $k = 3$. Ensuite nous examinons quelques variations du problème en imposant des contraintes supplémentaires aux sous-ensembles E^1 et E^2 comme par exemple qu'ils doivent former un arbre ou une collection de chaînes disjointes. Dans les deux cas nous présentons des conditions nécessaires et suffisantes pour qu'une solution existe. Nous considérons également le cas où le graphe est un graphe complet au lieu

d'un graphe biparti complet. Nous montrons que le problème de partitionnement dans un graphe complet est au moins aussi difficile que dans un graphe biparti complet. Dans le cas d'un graphe complet nous donnons également des conditions nécessaires et suffisantes pour qu'une solution existe si $E^1 \cup E^2$ est un arbre ou un cycle hamiltonien. Finalement nous considérons le cas où chacun des sous-ensembles E^1 et E^2 est structuré (deux chaînes hamiltoniennes disjointes, deux cycles disjoint par les arêtes) et nous donnons des conditions nécessaires et suffisantes pour l'existence d'une solution dans le cas d'un graphe complet et dans le cas d'un graphe biparti complet.

Mots-clés : Coloration de graphes, Graphe mixte, Ordonnancement, Tomographie discrète, Partitionnement d'arêtes, Complexité, Graphe biparti

Contents

Introduction	1
1 Preliminary definitions and results	5
1.1 Computational complexity	5
1.2 Graph theory	7
1.2.1 Basic definitions	7
1.2.2 Vertex coloring	11
1.2.3 Edge coloring and matchings	14
1.3 Discrete tomography	16
2 On two coloring problems in mixed graphs	21
2.1 Preliminaries	22
2.2 Strong mixed graph coloring problem	23
2.2.1 Bounds on the strong mixed chromatic number	23
2.2.2 Complexity results	25
2.3 Weak mixed graph coloring problem	34
2.3.1 Bounds on the weak mixed chromatic number	34
2.3.2 Complexity results	37
2.4 Conclusion	39
3 Bicolored matchings in some classes of graphs	41
3.1 Regular bipartite graphs	42
3.2 The interval property (IP)	48
3.3 Conclusion	51
4 On a graph coloring problem arising from discrete tomography	53
4.1 Preliminaries	55
4.2 Arbitrary colorings	55

4.2.1	Difficult problems for $k = 2$	56
4.2.2	Polynomially solvable cases with $k = 2$	59
4.2.3	The case $k \geq 3$	61
4.2.4	The case where G is a chain or a tree and $k > 2$	62
4.3	Proper colorings	65
4.3.1	Solvable cases of proper colorings	65
4.3.2	Difficult cases of proper colorings	69
4.4	Edge colorings	70
4.4.1	Arbitrary colorings	70
4.4.2	Proper colorings	71
4.5	Summary and conclusion	73
5	Graph coloring with cardinality constraints on the neighborhoods	77
5.1	\mathcal{NP} -completeness results	79
5.2	The special case of trees	84
5.3	The case of $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$	86
5.4	Conclusion	89
6	Degree-constrained edge partitioning in graphs arising from discrete tomography	91
6.1	The case where E^{12} is a tree in $K_{X,Y}$	93
6.2	The case where E^{12} is a collection of vertex disjoint chains in $K_{X,Y}$	96
6.3	The case where G is a complete graph K_X	97
6.4	A short incursion in the field of ‘oriented’ discrete tomography	98
6.5	Sufficient conditions for a solution of the image reconstruction problem with $k = 3$ colors	100
6.6	Cases where each one of E^1, E^2 is structured	101
6.7	Conclusion	106
	Conclusion	107
	Bibliography	113
	Index	119
	Curriculum vitae	122

Introduction

Many real world problems are easy to state, easy to explain and even easy to understand but they are very hard to solve, which means that even with the help of computers one cannot find an optimal solution in a reasonable amount of time. It would take years, decades or even centuries to get an optimal solution for these problems.

Combinatorial optimization deals with problems which have a finite number (or at least a countable number) of solutions. This does not make them easier to solve because the number of solutions may be exponentially high. Thus even for these problems, looking for an optimal solution doesn't seem to be the best approach. Combinatorial optimization problems occur in a large variety of industries such as telecommunication, scheduling, transportation or inventory management. Of course not all problems are hard to solve. For some of them there are algorithms producing an optimal solution in polynomial time.

So it seems to be natural to first analyze a problem from a computational complexity point of view, which means that one has to check whether the considered problem is easy to solve (i.e., there exists a polynomial time algorithm which produces an optimal solution) or not. In the first case, the goal may be to develop algorithms solving the problem within a computing time which is as short as possible. In the second case, one may apply different methods. One consists in considering the same problem but under specific constraints and checking whether it remains hard or not. This allows a better understanding of what may make the problem hard to solve, and to detect the border between hard problems and easy problems. Another approach would be to develop algorithms which run in polynomial time but do not necessarily find an optimal solution. These may be *heuristics* which give a 'good' solution fast, i.e., the computing time of these algorithms is reasonable but there is no guarantee of optimality of the solution, or *approximation algorithms* which guarantee a certain performance, i.e., we are sure that the solution found is 'close' to the optimal solution. Since we do not treat these last two approaches in this thesis, we will not go into further details. The reader interested in this subject is referred to [73] and [77].

Real world problems, before being analyzed as combinatorial optimization problems, have first to be modelled. A powerful tool to do this is *graph theory*. Its origin goes back to the year 1736 when Leonhard Euler published a paper on the 'Seven Bridges of Königsberg problem' [30]. Nowadays graph theory has become one of the fastest growing areas of modern mathematics, one reason being the large range of applications that it has in such

diverse disciplines as computer science, chemistry, social science and resource planning.

One of the most famous problems in graph theory is probably the *graph coloring problem* which consists in associating a color (or integer) from a finite set $\{0, 1, \dots, k - 1\}$ to each vertex of a graph such that two vertices which are adjacent (i.e., linked by an edge) get different colors. In this thesis we will consider several variations of this basic coloring problem which arise from scheduling and discrete tomography. We shall essentially analyze these problems, which are hard to solve in general, using the first method described here above, namely considering special cases, and trying to determine whether the problems remain hard to solve or not.

We consider scheduling problems containing disjunctive constraints as well as precedence constraints. Disjunctive constraints impose that some pairs of jobs cannot be done simultaneously, and precedence constraints impose that some pairs of jobs must be executed in a specified order. These problems have not been paid much attention in the literature. They may be modelled by a mixed graph $G_M = (V, U, E)$, and a coloring c of the vertices of G_M under the given constraints provides a feasible schedule. In Chapter 2 we analyze this problem in two different versions and show that they are \mathcal{NP} -complete even in special classes of graphs. We also present some polynomially solvable cases and give upper bounds for the mixed chromatic number.

Scheduling is probably the most famous application of graph coloring. But there are other fields of applications where graph coloring and its variations are used, for example in tomography. Discrete tomography deals with the reconstruction of objects from their projections. We introduce the basic image reconstruction problem which consists in reconstructing a two dimensional image ($m \times n$ pixels) knowing the number of pixels of each color in each row and each column. This problem is easy if we have only two colors but it is \mathcal{NP} -complete if the number of colors is four. The case of three colors is still open. We present two different graph theoretical approaches for the basic image reconstruction problem. One consists in coloring, or more precisely in partitioning the vertex set of a grid graph with respect to some constraints which we will explain later. The other approach consists in partitioning the edge set of a complete bipartite graph with respect to some constraints which will be precised. In this thesis we will consider coloring (or partitioning) problems which are closely related to these two approaches. Complexity results will be given including also some polynomially solvable cases.

This thesis is organized as follows. In Chapter 1 we present an introduction to computational complexity as well as some basic concepts concerning graph theory and discrete tomography. Each one of Chapters 2 - 6 is based on some published or accepted or submitted paper. Chapter 2 (see [62, 63]) deals with two coloring problems in mixed graphs related to some scheduling problems. In Chapter 3 (see [20]) we consider a problem consisting in finding a minimum number of edges in a graph such that each maximum matching contains p edges of this set, for several values of p . Although it is not directly related to scheduling or discrete tomography, we present it for its graph theoretical interest. Finally Chapters 4, 5 and 6

deal with problems arising from discrete tomography. We consider some problems related to (Chapter 5 (see [6]) and Chapter 6 (see [7])) or generalizing (Chapter 4 (see [21])) the basic image reconstruction problem.

Chapter 1

Preliminary definitions and results

1.1 Computational complexity

A *problem* is a general question to be answered, usually possessing several *parameters* (also called *free variables*) whose values are left unspecified. In order to describe a problem, one has to give a description of its parameters as well as the properties that the *answer* (or *solution*) is required to satisfy. We distinguish two types of problems: *decision problems* are problems which have as answer ‘yes’ or ‘no’, and *optimization problems* are problems which have as answer an optimal value, i.e., a maximum or minimum value which is called the *optimal solution*.

An *instance* of a problem P , denoted by I_P , is obtained by specifying the values for all the parameters of the problem.

Algorithms are general step-by-step procedures used to find an answer to a given problem. An algorithm is said to *solve* a problem P if it can be applied to any instance I_P of P and always produce a solution for that instance. For a given problem P , we are in general interested in finding an algorithm able to solve that problem as fast as possible. The time requirements of an algorithm are normally expressed in terms of the *size* of an instance I_P , which is conveniently a measure of the amount of input data. For each possible size, the *time complexity function* of an algorithm expresses the largest amount of time needed by the algorithm to solve a problem instance of that size.

We say that a function $f(n)$ is $O(g(n))$ if there exists a constant c such that $|f(n)| \leq c|g(n)|$ for all possible values of n . An algorithm is called *polynomial time algorithm* if its time complexity function is $O(p(n))$ for some polynomial function p , where n denotes the input length. Thus we say that a problem P can be *solved in polynomial time* if there exists a polynomial time algorithm A which solves P .

Let us now focus on decision problems. We denote by $O(P)$ the set of instances I_P of a decision problem P which have answer ‘yes’.

Definition 1.1 (set \mathcal{P}). \mathcal{P} is the set of all decision problems which can be solved in polynomial time.

Definition 1.2 (set \mathcal{NP}). \mathcal{NP} is the set of all decision problems P such that for each instance $I_P \in O(P)$ there exists a proof verifiable in polynomial time that $I_P \in O(P)$.

We clearly have $\mathcal{P} \subseteq \mathcal{NP}$. In fact, suppose that $P \in \mathcal{P}$. If there is a polynomial time algorithm A solving P , then for each instance $I_P \in O(P)$, we can verify if $I_P \in O(P)$ by applying algorithm A which will give answer ‘yes’ in polynomial time. Whether we also have $\mathcal{NP} \subseteq \mathcal{P}$, and thus $\mathcal{P} = \mathcal{NP}$, is not known so far.

Among the decision problems in \mathcal{NP} , there are some which are classified as the ‘hardest’. In order to explain this, we need the following definition.

Definition 1.3 (polynomial time reduction). Let P_1 and P_2 be two problems. A polynomial time reduction from P_1 to P_2 is an algorithm A which solves P_1 by using an algorithm B solving P_2 such that if B was a polynomial time algorithm, then A would be a polynomial time algorithm.

If there is a polynomial time reduction from P_1 to P_2 , we denote it by $P_1 \leq^p P_2$. If algorithm B is used exactly once during the reduction, we say that there is a *polynomial time transformation* from P_1 to P_2 and we denote it by $P_1 \propto P_2$. Now we can define the ‘hardest problems in \mathcal{NP} ’.

Definition 1.4 (set \mathcal{NP} -complete). \mathcal{NP} -complete is the set of all decision problems P such that

- (i) $P \in \mathcal{NP}$;
- (ii) $P' \propto P, \forall P' \in \mathcal{NP}$.

If any single problem which is in \mathcal{NP} -complete can be solved in polynomial time, then all problems in \mathcal{NP} can be solved in polynomial time. Therefore a problem P in \mathcal{NP} has the property that if $\mathcal{P} \neq \mathcal{NP}$, then $P \in \mathcal{NP} - \mathcal{P}$, or more precisely, $P \in \mathcal{P}$ if and only if $\mathcal{P} = \mathcal{NP}$.

We defined here the notion of \mathcal{NP} -complete as being a set of decision problems having certain properties. In this thesis we will also use \mathcal{NP} -complete as an adjective, i.e., a problem P in \mathcal{NP} -complete will also be said *to be \mathcal{NP} -complete*.

Lots of results in this thesis concern the proof of \mathcal{NP} -completeness of some specified problem. In order to show that a given problem P is \mathcal{NP} -complete, we have to proceed as follows:

- (a) show that $P \in \mathcal{NP}$;
- (b) select a problem P' which is known to be \mathcal{NP} -complete;
- (c) show that $P' \propto P$.

In general it is easy to verify that a given problem P is in \mathcal{NP} . Therefore whenever we want to show that a problem is \mathcal{NP} -complete, we shall simply skip (a), and concentrate on (b) and (c).

The foundations of the theory of \mathcal{NP} -completeness were given by S. Cook in 1971 (see [17]). He proved that a particular problem, the ‘satisfiability problem’, has the property that every other problem in \mathcal{NP} can be polynomially transformed to it. So in other words he found the first \mathcal{NP} -complete problem.

In [36] and [48], further developments in the theory of \mathcal{NP} -completeness can be found as well as a list of \mathcal{NP} -complete problems.

Let us finally introduce the notion of \mathcal{NP} -hardness.

Definition 1.5 (set \mathcal{NP} -hard). \mathcal{NP} -hard is the set of all problems P such that there exists $P' \in \mathcal{NP}$ -complete with $P' \leq^p P$.

In particular each problem in \mathcal{NP} -complete is necessarily in \mathcal{NP} -hard. Furthermore, an optimization problem whose decision version is \mathcal{NP} -complete is in \mathcal{NP} -hard. This is obvious since the optimization problem could be used to solve the decision problem. Notice that a problem in \mathcal{NP} -hard is neither necessarily a decision problem nor a problem in \mathcal{NP} .

We may use the term of \mathcal{NP} -hard also as an adjective in this thesis.

1.2 Graph theory

1.2.1 Basic definitions

First we give some basic definitions concerning graph theory. More specific definitions will be defined in the concerned chapters. If even more information about graphs are needed, the reader is referred to [9].

Definition 1.6 (undirected graph). An undirected graph G is a pair (V, E) , where $V = \{v_1, \dots, v_n\}$ is a set of vertices and $E \subseteq \{[v_i, v_j] \mid v_i, v_j \in V\}$ is a set of unordered pairs of elements of V .

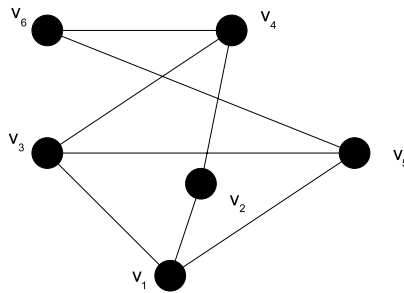


Figure 1.1: An undirected graph $G = (V, E)$ with $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $E = \{[v_1, v_2], [v_1, v_3], [v_1, v_5], [v_2, v_4], [v_3, v_4], [v_3, v_5], [v_4, v_6], [v_5, v_6]\}$.

Definition 1.7 (directed graph). A directed graph G is a pair (V, U) , where $V = \{v_1, \dots, v_n\}$ is a set of vertices and $U \subseteq \{(v_i, v_j) \mid v_i, v_j \in V\}$ is a set of ordered pairs of elements of V .

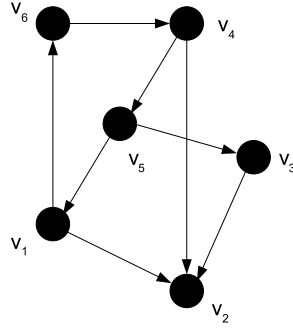


Figure 1.2: A directed graph $G = (V, U)$ with $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $U = \{(v_1, v_2), (v_1, v_6), (v_5, v_1), (v_3, v_2), (v_4, v_2), (v_5, v_3), (v_4, v_5), (v_6, v_4)\}$.

Definition 1.8 (mixed graph). A mixed graph G_M is a triple (V, U, E) , where $V = \{v_1, \dots, v_n\}$ is a set of vertices, $U \subseteq \{(v_i, v_j) \mid v_i, v_j \in V\}$ is a set of ordered pairs of elements of V , and $E \subseteq \{[v_i, v_j] \mid v_i, v_j \in V\}$ is a set of unordered pairs of elements of V .

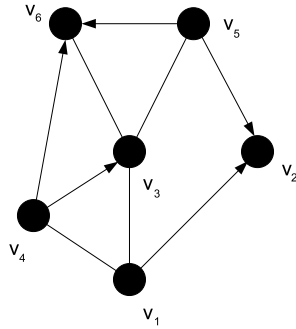


Figure 1.3: A mixed graph $G_M = (V, U, E)$ with $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $U = \{(v_1, v_2), (v_5, v_2), (v_4, v_3), (v_4, v_6), (v_5, v_6)\}$ and $E = \{[v_1, v_3], [v_1, v_4], [v_3, v_5], [v_3, v_6]\}$.

The elements in V , called *vertices*, will be denoted in general by v_i , where $i = 1, 2, \dots, n = |V|$. The elements of U , called *arcs*, will be denoted by (v_i, v_j) , where $i, j \in \{1, 2, \dots, n\}$. If there is an arc (v_i, v_j) , we say that vertices v_i and v_j are *joined by an arc*, with v_i being the *tail* and v_j being the *head* of this arc. They are also called the *endvertices* of the arc. The elements of E , called *edges*, will be denoted by $[v_i, v_j]$, where $i, j \in \{1, 2, \dots, n\}$. If there is an edge $[v_i, v_j]$, we say that vertices v_i and v_j are *joined by an edge* and that they are the *endvertices* of that edge. Whenever we say that two vertices v_i and v_j are *adjacent*, we mean that there is an edge or/and an arc joining these vertices. Two edges (or two arcs, or one edge and one arc) having at least one common endvertex are also said to be *adjacent*. Here we will only consider graphs without loops, i.e., without edges $[v_i, v_i]$ and arcs (v_j, v_j) , where $i, j \in \{1, \dots, n\}$.

Notice that undirected and directed graphs are just special cases of mixed graphs (for undirected graphs $U = \emptyset$ and for directed graphs $E = \emptyset$). The number of arcs and edges

having a common endvertex v_i is called the *degree* of vertex v_i and will be denoted by $d(v_i)$. The *degree of a graph* $G_M = (V, U, E)$ is the maximum over all degrees of its vertices. We denote it by $\Delta(G_M)$, and thus $\Delta(G_M) = \max_{v_i \in V}(d(v_i))$. We also define the *indegree* of a vertex v_i as the number of arcs in G_M having v_i as head. It will be denoted by $d^-(v_i)$. Similarly we define the *outdegree* of a vertex v_i , denoted by $d^+(v_i)$, as the number of arcs in G_M having v_i as tail.

The following definitions are usually given for undirected graphs. We will do the same, but remark that one could easily adapt the definitions to the more general case of mixed graphs. Consider an undirected graph $G = (V, E)$.

Definition 1.9 (induced subgraph). *Let $A \subseteq V$ be a subset of the vertex set V . The subgraph of G induced by A is the graph $G_A = (A, E_A)$ where $E_A = \{[v_i, v_j] \in E \mid v_i, v_j \in A\}$.*

Definition 1.10 (partial graph). *$H = (V, E')$ is a partial graph of G if $E' \subseteq E$.*

Figure 1.4 shows an induced subgraph $G_A = (A, E_A)$ of $G = (V, E)$ from Figure 1.1 with $A = \{v_2, v_3, v_4, v_5, v_6\}$ and a partial graph $H = (V, E')$ of $G = (V, E)$ with $E' = \{[v_2, v_4], [v_3, v_4], [v_3, v_5], [v_4, v_6], [v_5, v_6]\}$.

Definition 1.11 (partial subgraph). *$H = (V', E')$ is a partial subgraph of G if it is an induced subgraph of a partial graph of G .*

An example of a partial subgraph is shown in Figure 1.5. It represents a subgraph induced by $V' = \{v_1, v_2, v_3, v_4, v_5\}$ of the partial graph H from Figure 1.4b).

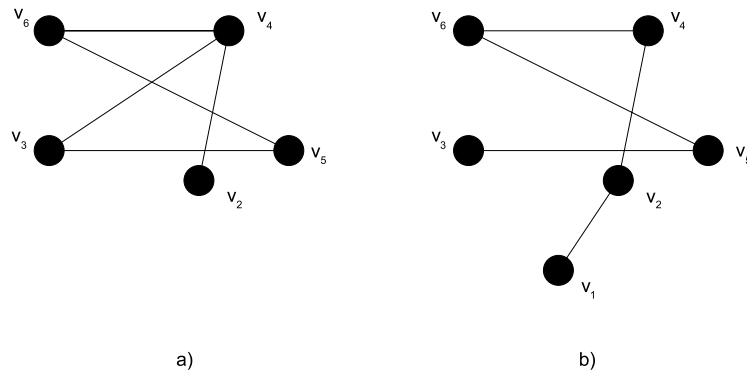


Figure 1.4: a) An induced subgraph $G_A = (A, E_A)$ of $G = (V, E)$. b) A partial graph $H = (V, E')$ of $G = (V, E)$.

The next two concepts play an important role in vertex coloring and edge coloring as we will see later in the next section.

Definition 1.12 (stable set). *A stable set is a set of pairwise nonadjacent vertices.*

Definition 1.13 (matching). *A matching is a set of pairwise nonadjacent edges.*

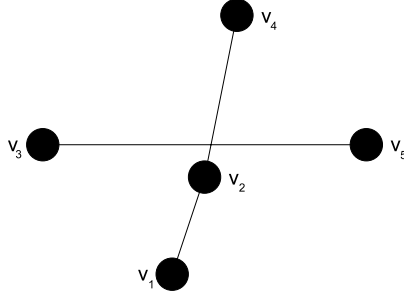


Figure 1.5: A partial subgraph $H = (V', E')$ of $G = (V, E)$.

For the following definitions, let us consider a mixed graph $G_M = (V, U, E)$ with $|V| = n$.

Definition 1.14 (chain). A chain is a sequence of vertices $\{v_1, v_2, \dots, v_l\}$ such that $[v_i, v_{i+1}] \in E$ or $(v_i, v_{i+1}) \in U$ or $(v_{i+1}, v_i) \in U$ for $i = 1, \dots, l - 1$.

Definition 1.15 (cycle). A cycle (or closed chain) is a sequence of vertices $\{v_1, v_2, \dots, v_l\}$ such that $[v_i, v_{i+1}] \in E$ or $(v_i, v_{i+1}) \in U$ or $(v_{i+1}, v_i) \in U$ for $i = 1, \dots, l - 1$ and $[v_l, v_1] \in E$ or $(v_l, v_1) \in U$ or $(v_1, v_l) \in U$.

Definition 1.16 (path). A path is a sequence of vertices $\{v_1, v_2, \dots, v_l\}$ such that $(v_i, v_{i+1}) \in U$ for $i = 1, \dots, l - 1$.

Definition 1.17 (circuit). A circuit is a sequence of vertices $\{v_1, v_2, \dots, v_l\}$ such that $(v_i, v_{i+1}) \in U$ for $i = 1, \dots, l - 1$ and $(v_l, v_1) \in U$.

Figure 1.6 shows examples of a chain, a cycle, a path and a circuit.

Notice that a path is a special case of a chain and a circuit is a special case of a cycle. The *length* of a chain (resp. cycle, path, circuit) is the number of edges and arcs contained in the chain (resp. cycle, path, circuit). A chain (resp. cycle, path, circuit) is said to be *elementary* if $v_i \neq v_j$ for all $i \neq j$, $i, j \in \{1, \dots, l\}$.

A chain (resp. cycle path, circuit) is called *Hamiltonian* if it contains all the vertices of the graph.

Definition 1.18 (connected graph). A mixed graph $G_M = (V, U, E)$ is said to be *connected* if for all pairs of vertices $v_i, v_j \in V$, there exists a chain joining these two vertices.

Definition 1.19 (strongly connected graph). A mixed graph $G_M = (V, U, E)$ is said to be *strongly connected* if for all pairs of vertices $v_i, v_j \in V$, there exists a path from v_i to v_j and from v_j to v_i .

Let us finally define some special classes of graphs which will be frequently used in this thesis.

Definition 1.20 (tree). A mixed graph $G_M = (V, U, E)$ is called a *tree* if it is connected and contains no cycle.

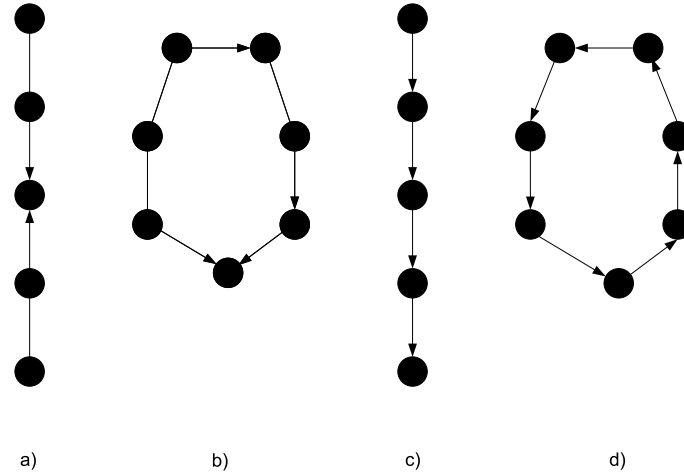


Figure 1.6: a) A chain. b) A cycle. c) A path. d) A circuit.

Definition 1.21 (Δ -regular graph). A mixed graph $G_M = (V, U, E)$ is a Δ -regular graph if $d(v_i) = \Delta$, for all $v_i \in V$.

Definition 1.22 (bipartite graph). A mixed graph $G_M = (V, U, E)$ is bipartite if its vertex set V can be partitioned into two stable sets V_1 and V_2 .

In this case we write $G_M = (V_1, V_2, U, E)$.

1.2.2 Vertex coloring

In this section we will introduce one of the most famous problems in graph theory and combinatorial optimization, namely the *graph coloring problem*. This problem has been studied intensively in the literature (see for instance [8, 47]).

Definition 1.23 (vertex coloring of G). Let $G = (V, E)$ be an undirected graph. A vertex coloring of G is a mapping $c: V \rightarrow C$, where C is the set of colors (typically the natural numbers), such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$.

If the number of available colors is k , i.e., if $|C| = k$, then c is called a k -coloring. Given a graph $G = (V, E)$, the smallest integer k such that there exists a k -coloring of the vertices of G is called the *chromatic number* of G and denoted by $\chi(G)$. The graph coloring problem in its decision version may be formulated as follows.

Graph Coloring Problem $GC(G, k)$

Instance: An undirected graph $G = (V, E)$ and an integer $k \geq 1$.

Question: Does G admit a k -coloring?

While the graph coloring problem is solvable in polynomial time for $k = 2$, it remains \mathcal{NP} -complete for all fixed $k \geq 3$ (see [48]).

Notice that coloring the vertices of a graph $G = (V, E)$ using k colors is equivalent to partition the vertex set V into k stable sets V_1, \dots, V_k . In fact, the set of vertices having the same color s form a stable set since they cannot be pairwise adjacent.

The graph coloring problem has a large range of applications (see for instance [57, 77]). We shall give here two examples of applications: timetabling and frequency assignment.

For the timetabling problem, suppose for instance that a set of courses has to be scheduled, i.e., one has to associate a time period with each course respecting the following disjunctive constraints: a class cannot have two different courses at the same time period and a teacher cannot give two different courses at the same time period. If there are k time periods available, we may want to find out whether it is possible to determine a timetabling of these courses respecting the constraints mentioned above. To represent this problem as a graph coloring problem, we associate with each course a vertex and two vertices are linked if the corresponding courses are given by the same teacher or if there is a class attending the two courses. If the resulting graph is k -colorable, then there exists such a timetabling and each color corresponds to a time period.

In the frequency assignment problem (FAP), a set of wireless communication connections (or a set of antennae) must be assigned frequencies such that data transmission between the two endpoints of each connection (the receivers) is possible. The frequencies assigned to two connections which are close to each other incur interference, resulting in quality loss of the signal if they are assigned the same frequency. The FAP consists in assigning frequencies to the connections such that there is no interference. Given k frequencies and the network of antennae, is it possible to assign a frequency to each antenna such that there is no interference? Again, this problem may be transformed into a graph coloring problem by associating a vertex to each antenna, and by joining two vertices if the corresponding antennae are close to each other and their frequencies may incur interference. If there exists a k -coloring of the resulting graph, then by associating each color to a frequency, we get a solution for the FAP.

But nevertheless, it may happen that some constraints cannot be modelled by simply using an undirected graph and hence the coloring model described above may be too limited to be useful in some applications. For example, if we want to include some precedence constraints in our timetabling problem (i.e., we impose for some pairs of courses that they must be scheduled in a given order), the model given by the undirected graph will not be sufficient anymore. In this case, we have to introduce a new model which will be given by mixed graphs.

Definition 1.24 (strong mixed graph coloring of G_M). *Let $G_M = (V, U, E)$ be a mixed graph. A strong mixed graph coloring of G_M is a mapping $c: V \rightarrow C$, where C is the set*

of colors (typically the natural numbers), such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$ and for each arc $(v_l, v_q) \in U$, $c(v_l) < c(v_q)$.

As before, if the number of available colors is k , i.e., if $|C| = k$, then c is called a *strong mixed graph k -coloring*. Given a mixed graph $G_M = (V, U, E)$, the smallest integer k , such that there exists a strong mixed graph k -coloring of the vertices of G_M , is called the *strong mixed chromatic number* of G_M and denoted by $\gamma_M(G_M)$.

The strong mixed graph coloring problem in its decision version may be stated as follows.

Strong Mixed Graph Coloring Problem $S(G_M, k)$

Instance: A mixed graph $G_M = (V, U, E)$ and an integer $k \geq 1$.

Question: Does G_M admit a strong mixed graph k -coloring?

It can easily be seen, that a strong mixed graph coloring of a mixed graph will give an answer to our timetabling problem containing precedence constraints. In fact, we join two vertices v_l and v_q by an arc (v_l, v_q) if the course corresponding to vertex v_l must be scheduled before the course corresponding to vertex v_q . Thus the resulting graph is a mixed graph and there is a one-to-one correspondance between a strong mixed graph k -coloring and a feasible schedule over k time periods.

Notice that for a strong mixed graph coloring to exist, the mixed graph must not contain any circuit. It is easy to see that the graph coloring problem is just a special case of the strong mixed graph coloring problem where we have no precedence constraints. Thus $S(G_M, k)$ is also \mathcal{NP} -complete for all fixed $k \geq 3$.

Suppose now that for the pairs of courses in our timetabling problem which have been linked by a precedence constraint, we decide that they can take place at the same time. In other words, we allow the pairs of vertices which are linked by an arc to get the same color. To take these relaxations into account, we have to introduce yet another type of mixed graph coloring.

Definition 1.25 (weak mixed graph coloring of G_M). Let $G_M = (V, U, E)$ be a mixed graph. A weak mixed graph coloring of G_M is a mapping $c : V \rightarrow C$, where C is the set of colors (typically the natural numbers), such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$, and for each arc $(v_l, v_q) \in U$, $c(v_l) \leq c(v_q)$.

Again, if the number of available colors is k , i.e., if $|C| = k$, then c is called a *weak mixed graph k -coloring*. Given a mixed graph $G_M = (V, U, E)$, the smallest integer k such that there exists a weak mixed graph k -coloring of the vertices of G_M is called the *weak mixed*

chromatic number of G_M and denoted by $\chi_M(G_M)$.

The weak mixed graph coloring problem in its decision version may be stated as follows.

Weak Mixed Graph Coloring Problem $W(G_M, k)$

Instance: A mixed graph $G_M = (V, U, E)$ and an integer $k \geq 1$.

Question: Does G_M admit a weak mixed graph k -coloring?

Clearly this type of mixed graph coloring may solve our new timetabling problem. It contains the strong mixed graph coloring problem in the sense that the latter one is just a special case of it. In fact, if in the weak mixed graph coloring problem we suppose that each pair of vertices linked by an arc is also linked by an edge, we again obtain the strong mixed graph coloring problem. Thus $W(G_M, k)$ is \mathcal{NP} -complete.

While the graph coloring problem has been studied intensively, the strong and weak mixed graph coloring problems have not found much attention in the literature. In Chapter 2 of this thesis, we investigate these two coloring problems. First we give an overview of the work that has already been done. Then for both problems, we determine some bounds on the mixed chromatic numbers. Finally we analyze some special classes of mixed graphs for which we show that either the problem considered is \mathcal{NP} -complete or polynomially solvable.

1.2.3 Edge coloring and matchings

In this section we present another famous coloring problem in undirected graphs: the *edge coloring problem*. As the vertex coloring problem, it has been studied intensively (see for instance [43, 74]) and it also has a large range of applications, especially in scheduling (see [27]).

Definition 1.26 (edge coloring). *Let $G = (V, E)$ be an undirected graph. An edge coloring of G is a mapping $e : E \rightarrow C$ such that for each pair of edges $[v_i, v_j], [v_i, v_l]$, $e([v_i, v_j]) \neq e([v_i, v_l])$.*

We call e an *edge k -coloring* if the number of colors available is k , i.e., if $|C| = k$. Given an undirected graph $G = (V, E)$, the smallest integer k , such that there exists an edge k -coloring of G , is called the *chromatic index* of G and denoted by $\chi'(G)$.

Hence the edge coloring problem may be defined as follows.

Edge Coloring Problem $EC(G, k)$

Instance: An undirected graph $G = (V, E)$ and an integer $k \geq 1$.

Question: Does G admit an edge k -coloring?

As the vertex coloring problem, the edge coloring problem $EC(G, k)$ is in general \mathcal{NP} -complete (see [43]). But there are some special classes of graphs for which polynomial time algorithms are known. A famous result concerning edge coloring is König's coloring theorem which can be stated as follows.

Theorem 1.27 (König's coloring theorem [4]). *Let $G = (V_1, V_2, E)$ be an undirected bipartite graph. Then $\chi'(G) = \Delta(G)$.*

Similar to the case of vertex coloring, the edge coloring problem may be seen as a partition problem. Indeed, the set of edges with a same color f must be pairwise non adjacent. So they form a matching. Thus, coloring the edges of a graph $G = (V, E)$ with k colors is equivalent to partition the edge set E into k sets E_1, \dots, E_k , each one being a matching.

The matching theory has been studied intensively and plays an important role since many discrete problems can be formulated as matching problems (e.g. timetabling problems). A matching M in a graph $G = (V, E)$ is called *maximum* if for all matchings M' in G , we have $|M| \geq |M'|$. A *perfect matching* M in a graph $G = (V, E)$ is a matching such that for each vertex $v_i \in V$, there exists an edge $[v_i, v_j] \in M$. In contrast to the vertex and edge coloring problems, the problem of finding a maximum matching in a general graph is polynomially solvable (see [28, 29]).

In this thesis, we will not consider the edge coloring problem or the maximum matching problem themselves, but we will concentrate on some problems which are in a certain way related to them. In Chapter 3 we consider the problem of characterizing a minimum set R for which maximum matchings M can be found with specific values of $p = |M \cap R|$. More precisely, given a graph $G = (V, E)$ and a set $\mathcal{P} = \{p_0, p_1, \dots, p_s\}$ of integers $0 \leq p_0 < p_1 < \dots < p_s \leq \lfloor |V|/2 \rfloor$, we want to color a subset $R \subseteq E$ of edges of G , say in red, in such a way that for any i ($0 \leq i \leq s$) G contains a maximum matching M_i with exactly p_i red edges, i.e., $|M_i \cap R| = p_i$. Notice that we do not impose that two red edges are necessarily non adjacent. So we consider a problem of edge partitioning rather than edge coloring in the way it was defined above.

Edge coloring, or more precisely edge partitioning, is also closely related to what is called *discrete tomography*. In the next section we will introduce this field and explain this relation.

1.3 Discrete tomography

Tomography deals with the reconstruction of an object from its projections in given directions. While these projections are uniquely defined, this is not necessarily the case for the underlying object or it may not exist at all. Thus once we are given the projections, we may be interested in the following problems.

- (1) Does there exist an object with the given projections?
- (2) Is there a unique object corresponding to the projections?
- (3) How can the object, if it exists, be reconstructed?

Tomography has its main application in medical imagery, where continuous 3-dimensional objects must be reconstructed from several projections, which consist in 2-dimensional functions. But there are lots of other areas in which tomography plays an important role, for example in crystallography, data compression, pattern recognition, electron microscopy or timetabling.

In *discrete tomography*, the object to be reconstructed can be expressed using a discrete set of data. One of the most famous problems in discrete tomography is the reconstruction problem of 2-dimensional images from two projections (horizontal and vertical), which we simply call *basic image reconstruction problem*. Here we consider an $(m \times n)$ array $A = (a_{ij})$ where each entry may contain a pixel to be colored with one of the colors $1, 2, \dots, k$. Given the number h_i^s (resp. v_j^s) of pixels with color s in row i (resp. column j) for $s = 1, \dots, k$ and $i = 1, \dots, m$ (resp. $j = 1, \dots, n$), the basic image reconstruction problem consists in assigning a color in $\{1, \dots, k\}$ to each entry of A such that there are exactly h_i^s (resp. v_j^s) pixels in row i (resp. column j) having color s , for $s = 1, \dots, k$ and $i = 1, \dots, m$ (resp. $j = 1, \dots, n$). Clearly the values h_i^s and v_j^s must satisfy some necessary conditions for a solution to exist:

$$\sum_{s=1}^k h_i^s = n \quad (i = 1, \dots, m) \quad (1.1)$$

$$\sum_{s=1}^k v_j^s = m \quad (j = 1, \dots, n) \quad (1.2)$$

$$\sum_{i=1}^m h_i^s = \sum_{j=1}^n v_j^s \quad (s = 1, \dots, k) \quad (1.3)$$

This basic image reconstruction problem, also called *colored matrix reconstruction problem*, occurs in problems associated with the location of atoms in a crystal by means of X-rays parallel to the two coordinate axes. There are k types of atoms present and each X-ray gives the number of atoms of each type lying on each line in a family of parallel lattice lines. Another application concerns timetabling (see [18]). With each row we associate a class, with each column a time period, and with each color a professor. Then the horizontal

projection h_i^s corresponds to the number of one-hour lectures that teacher s must give to class i , and the vertical projection v_j^s is equal to 1 if teacher s is available during period j and 0 else. A feasible timetable is an assignment of colors to the entries with respect to the projections.

In the case of two colors, i.e., $k = 2$, the basic image reconstruction problem corresponds to the reconstruction of a binary matrix, given the number of occurrences of 1's in each row and each column. This case has been intensively studied (see for instance [65]). In 1957, Ryser and Gale (independently from each other) gave necessary and sufficient conditions under which there exists an image satisfying the projections. They may be formulated as follows.

Consider the matrix \bar{A} in which each row i consists of h_i^1 consecutive 1's followed by $n - h_i^1$ 0's. Such a matrix \bar{A} is uniquely determined by its row sum vector. Let us denote by \bar{Q} the column sum vector of A . Furthermore, let us put the values h_i^1 (resp. v_j^1) in nonincreasing order and denote them by $p_1 \geq p_2 \geq \dots \geq p_m$ (resp. $q_1 \geq q_2 \geq \dots \geq q_n$). Then the Ryser conditions are $\sum_{j=l}^n q_j \geq \sum_{j=l}^n \bar{Q}_j$, for $2 \leq l \leq n$. Since these conditions may be verified in polynomial time, the image reconstruction problem for two colors can be solved in polynomial time. Besides this, the proof also provides an algorithm which reconstructs the corresponding image (if it exists) in time $O(n(m + \log(n)))$. In the case of two colors, the problem of uniqueness of a solution has also been solved by Ryser. He shows that a binary matrix A is nonunique if and only if it contains a switching component which consists of a 2×2 submatrix having two 1's on one diagonal and two 0's on the other one.

In 2002, Chrobak and Dürr showed that the basic image reconstruction problem with $k = 4$ colors is \mathcal{NP} -complete (see [16]). The complexity is still open for $k = 3$. In [18, 19] some polynomial solvable cases are given. In [18] it is shown for instance that the basic image reconstruction problem with $k = 3$ colors is polynomial solvable, if for colors 1 and 2 we have $h_i^1, h_i^2, v_j^1, v_j^2 \in \{0, 2\}$ for each row i and each column j . In [19] the case of $k = 3$ colors is considered with the restriction that one color, say color 1, may have several occurrences, but not more than r in a fixed number q of rows and columns, and it is shown to be polynomially solvable.

For a more complete overview of the theory and the applications of discrete tomography, the reader is referred to [52, 53].

We will now show how the basic image reconstruction problem may be formulated in graph theoretical terms. This formulation has been introduced for the first time in [18] and it has been used to show several complexity results in [18, 19]. With each row i of A we associate a vertex i . We denote by X the set of these vertices. Similarly we associate a vertex j with each column j of A and denote by Y the set of these vertices. Then with each entry

a_{ij} , we associate an edge $[i, j]$. Hence we get a complete bipartite graph $K_{X,Y}$. The basic image reconstruction problem then consists in finding a partition E^1, \dots, E^k of the edge set of $K_{X,Y}$ such that for each vertex $i \in X$, there are exactly h_i^s edges of E^s which are incident to i , and for each vertex $j \in Y$, there are exactly v_j^s edges of E^s which are incident to j , for $s = 1, \dots, k$. Thus the basic image reconstruction problem may be interpreted as a special case of edge coloring where we allow adjacent edges to have the same color, but we impose the number of edges of each color incident to each vertex of the graph.

But the basic image reconstruction problem may also be seen as a special kind of vertex coloring. Indeed, let us associate a vertex with each pixel in the array A . We join two vertices if the corresponding pixels are adjacent in A ; we obtain a graph G which is called *grid graph*. Notice that each column in A as well as each row in A corresponds to a chain in the graph G . Since we want to associate a color with each pixel by respecting the given projections (number of occurrences of each color in each row and each column), we equivalently want to color the vertices of G such that in each chain corresponding to a column or to a row, we have a given number of occurrences of each color. Notice that we do not care about adjacent vertices having the same color. Thus we get a vertex coloring problem which is equivalent to the basic image reconstruction problem.

In Chapter 4, we will concentrate on a generalization of this particular vertex coloring problem. We consider an undirected graph $G = (V, E)$ and a collection \mathcal{P} of chains covering the vertices of G . We are given a set of colors $\{1, 2, \dots, k\}$ as well as for each chain P_i the number of occurrences of each color j denoted by h_i^j , for $j = 1, 2, \dots, k$. We have to find a partition V^1, V^2, \dots, V^k of V such that for each chain P_i we have $|P_i \cap V^j| = h_i^j$, for $j = 1, 2, \dots, k$. This problem will be referred to as $\Lambda(G, k, \mathcal{P}, H)$, where H is the collection of vectors $h(P_i) = (h_i^1, \dots, h_i^k)$. We may analyze the same problem with the additional requirement that two adjacent vertices do not get the same color, i.e., we want to get a graph coloring as defined previously. This problem will also be studied in Chapter 4 and we denote it by $\Lambda^*(G, k, \mathcal{P}, H)$. For both problems, we also consider the edge partitioning version, i.e., we want to find a partition E^1, E^2, \dots, E^k of E such that each chain contains a given number of edges of each color, and we may require or not that two adjacent edges do not belong to the same set of the partition. These problems will be respectively denoted by $\Psi(G, k, \mathcal{P}, H)$ and $\Psi^*(G, k, \mathcal{P}, H)$.

Instead of fixing the number of occurrences of each color in chains, one may be interested in coloring the vertices of an undirected graph $G = (V, E)$ by fixing the number of occurrences of each color in some generalized neighborhood of each vertex v . Let us denote by $N_d(v)$ the d -neighborhood of a vertex v which is the set of vertices which are at distance d from v (i.e., the shortest chain between each one of these vertices and v has length d). Given an undirected graph $G = (V, E)$, a set of colors $\{1, \dots, k\}$ and a collection H of vectors $h(v) = (h_v^1, \dots, h_v^k)$, $v \in V$, we want to find a partition V^1, \dots, V^k of V such that

$|N_d(v) \cap V^i| = h_v^i$ for all vertices $v \in V$ and all $1 \leq i \leq k$. This problem is referred to as $\mathcal{P}_d(G, H, k)$. One may also define $N_d^+(v)$ as the set of vertices which are at distance at most d from v . The corresponding partition problem will be denoted by $\mathcal{P}_d^+(G, H, k)$. As before we can also analyze these problems with the additional requirement that the coloring must be proper, i.e., two adjacent vertices must belong to different sets $V^i, V^j, i \neq j$. Then the corresponding partition problems will be denoted by $\mathcal{P}_d^*(G, H, k)$ and $\mathcal{P}_d^{+*}(G, H, k)$, respectively. In Chapter 5, we deal with the cases $d = 1$ and $d = 2$ for some values of k by determining the complexity of the different problems. Finally we will also be interested in the case where we do not fix the number of occurrences of each color in some generalized neighborhood, but we simply give an upper bound on this number. In other words, we want to find a partition V^1, \dots, V^k of V such that $|N_d(v) \cap V^i| \leq h_v^i$ for all vertices $v \in V$ and all $1 \leq i \leq k$. Results for $d = 1$ are given in Chapter 5, for the case of non proper coloring as well as for the case of proper coloring. These problems will be denoted by $\mathcal{BP}(G, H, k)$ and $\mathcal{BP}^*(G, H, k)$, respectively.

As already mentioned before, the complexity status of the basic image reconstruction problem with $k = 3$ colors is still open. Some special cases have been shown to be polynomially solvable in [18, 19]. In Chapter 6, we use the edge partitioning version of the basic image reconstruction problem, and we present another special case for which this problem can be solved in polynomial time. Furthermore we consider the case of $k = 3$ colors where the sets E^1 and E^2 have to satisfy some additional constraints. More precisely we consider the cases where $E^1 \cup E^2$ must form a tree or a collection of disjoint chains. We give necessary and sufficient conditions for a solution to exist. These conditions can be checked in polynomial time, and the proofs provide polynomial time algorithms to construct a solution if there is one. We also consider the case where each set E^1 and E^2 has to be a Hamiltonian chain, as well as the case where they have to be edge-disjoint cycles through specified vertices. Again, we give necessary and sufficient conditions for a solution to exist. Notice that these cases do not correspond to special cases of the basic image reconstruction problem with $k = 3$ colors, since we impose some additional constraints on the sets E^1 and E^2 .

Although there is no immediate connection with discrete tomography, we do consider the case where the graph G is simply a complete graph K_X instead of a complete bipartite graph. In Chapter 6 we give results for the same problems treated in the case of $K_{X,Y}$.

Finally we introduce in Chapter 6 the field of ‘oriented’ discrete tomography. Here we consider a complete symmetric oriented graph $G = \vec{K}_X$. For each vertex i of X we are given the number of incoming (h_i^{-s}) and outgoing (h_i^{+s}) arcs of each color s . Then the basic image reconstruction problem in its oriented version is as follows. We want to find a partition $\vec{E}^1, \dots, \vec{E}^k$ of the arc set \vec{E} such that for each color s and for each vertex i , we have exactly h_i^{-s} incoming arcs belonging to \vec{E}^s and h_i^{+s} outgoing arcs belonging to \vec{E}^s . We will consider the case of $k = 3$ colors, and we give necessary and sufficient conditions for a solution to exist in the case where $\vec{E}^1 \cup \vec{E}^2$ has to be a tree.

Chapter 2

On two coloring problems in mixed graphs

Introduction

A mixed graph $G_M = (V, U, E)$ on vertex set $V = \{v_1, v_2, \dots, v_n\}$ is a graph containing arcs (set U) and edges (set E). We denote by $[v_i, v_j]$ an edge joining vertices v_i and v_j and by (v_l, v_q) an arc oriented from v_l to v_q . Here we consider only connected finite mixed graphs containing no multiple edges, no multiple arcs, and no loops. The number of vertices in a mixed graph $G_M = (V, U, E)$ will be denoted by $|V| = n$. Mixed graphs have been introduced for the first time in [69].

In this chapter we are interested in two coloring problems in mixed graphs, which have already been presented in Chapter 1 in relation with some scheduling problems. Let us recall some definitions and notations. The first problem is called *strong mixed graph coloring problem*. A *strong mixed graph k -coloring* of a mixed graph G_M is a mapping $c: V \rightarrow \{0, 1, \dots, k-1\}$ such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$, and for each arc $(v_l, v_q) \in U$, $c(v_l) < c(v_q)$. Notice that such a coloring can exist if and only if the mixed graph G_M does not contain any directed circuit. We denote by $\gamma_M(G_M)$ the *strong mixed chromatic number* of G_M , that is the smallest integer k such that G_M admits a strong mixed graph k -coloring. A mixed coloring of G_M with $\gamma_M(G_M)$ colors will be called *optimal*. We will generally consider the following problem. Given a mixed graph $G_M = (V, U, E)$ and a positive integer k , find out whether G_M admits a strong mixed graph k -coloring. This coloring problem has been studied in [33, 41, 67, 68]. In [41] some upper bounds on the strong mixed chromatic number are given. An $O(n^2)$ time algorithm to color optimally mixed trees and a branch-and-bound algorithm are also developed. In [33] a linear time algorithm for mixed trees is given as well as an $O(n^{3.376} \log(n))$ time algorithm for series parallel graphs. Finally in [67, 68], the unit-time job-shop problem is considered via strong mixed graph coloring. In this particular case the partial graph (V, \emptyset, E) is a disjoint union

of cliques and the undirected graph (V, U, \emptyset) is a disjoint union of directed paths. In [68] three branch-and-bound algorithms are presented and tested on randomly generated mixed graphs of order $n \leq 200$ for an exact solution and of order $n \leq 900$ for an approximate solution. Also some complexity results are given concerning this special class of mixed graphs.

The second problem which we will consider in this chapter is called the *weak mixed graph coloring problem* which was introduced for the first time in [69]. A *weak mixed graph k -coloring* of a mixed graph G_M is a mapping $c: V \rightarrow \{0, 1, \dots, k-1\}$ such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$, and for each arc $(v_l, v_q) \in U$, $c(v_l) \leq c(v_q)$. Notice that in such a coloring of a mixed graph, all vertices on a directed circuit must necessarily have the same color. We denote by $\chi_M(G_M)$ the *weak mixed chromatic number* of G_M , that is the smallest integer k such that G_M admits a weak mixed k -coloring. Given a mixed graph $G_M = (V, U, E)$ and a positive integer k , we are interested in finding out whether G_M admits a weak mixed graph k -coloring. The weak mixed graph coloring problem has been studied in [2, 50, 54, 69, 70]. In [70] some algorithms calculating the exact value of the weak mixed chromatic number of graphs of order $n \leq 40$ and upper bounds for graphs of order larger than 40 are presented.

This chapter is organized as follows. In Section 2.1 we give some definitions and notations which will be used later. Section 2.2 deals with the strong mixed graph coloring problem. Some bounds on the strong mixed chromatic number are given, as well as some complexity results concerning special classes of graphs. In Section 2.3 the weak mixed graph coloring problem is considered and bounds on the weak mixed chromatic number are given with some complexity results.

2.1 Preliminaries

Let $G_M = (V, U, E)$ be a mixed graph and let V_o be the set of vertices which are incident to at least one arc in G_M . We denote by $G(V_o)$ the mixed subgraph of G_M induced by V_o and by $G_M^o = (V_o, U, \emptyset)$ the directed partial graph of $G(V_o)$. $n(G_M^o)$ denotes the number of vertices on a longest directed path in G_M^o . Notice that the length of a longest directed path in G_M (i.e., the number of arcs of a longest directed path in G_M) is equal to $n(G_M^o) - 1$. Clearly $n(G_M^o)$ is a lower bound on $\gamma_M(G_M)$.

Let P be a directed path in G_M^o . The number of vertices in P will be denoted by $|P|$.

Let v_i be a vertex in G_M . The *inrank* of v_i , denoted by $in(v_i)$, is the length of a longest directed path in G_M^o ending at vertex v_i . Similarly we define the *outrank* of v_i , denoted by $out(v_i)$, as being the length of a longest directed path in G_M^o starting at vertex v_i . If v_i is not incident to any arc, then $in(v_i) = out(v_i) = 0$. Notice that the length of a longest directed path in G_M is given by $\max_{v_i \in V} (in(v_i) + out(v_i))$.

Notice that the parameters introduced above can only be defined if G_M^o has no directed circuit.

We recall that the *degree* of a vertex v in G_M , denoted by $d_{G_M}(v)$, is the number of edges and arcs incident to v . We shall simply write $d(v)$ if no confusion can occur.

2.2 Strong mixed graph coloring problem

In this section we study the following problem, which is called the *strong mixed graph coloring problem* and has already been introduced in Chapter 1.

Instance: A mixed graph $G_M = (V, U, E)$ and an integer $k \geq n(G_M^o)$.

Question: Can the vertices of G_M be colored using at most k colors such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$, and for each arc $(v_l, v_q) \in U$, $c(v_l) < c(v_q)$?

We will refer to this problem as $S(G_M, k)$. Notice that in this problem we can suppose w.l.o.g. that whenever $(v_l, v_q) \in U$, then $[v_l, v_q] \notin E$ since $(v_l, v_q) \in U$ implies that $c(v_l) < c(v_q)$ and thus $c(v_l) \neq c(v_q)$.

A necessary and sufficient condition for a mixed graph to admit a strong mixed graph coloring is that it does not contain any directed circuit. We will suppose for the rest of this section that it is satisfied.

2.2.1 Bounds on the strong mixed chromatic number

Upper bounds on the mixed chromatic number have been given in [41]. In particular, one of these bounds implies that for mixed bipartite graphs we have $n(G_M^o) \leq \gamma_M(G_M) \leq n(G_M^o) + 1$. In this section we will give some upper bounds for special classes of mixed graphs and in some cases the exact value of the strong mixed chromatic number.

Lemma 2.1. *Let $G_M^o = (V_1 \cup V_2, U, \emptyset)$ be a mixed bipartite graph. Assume that all paths of length $n(G_M^o) - 1$ start in the same vertex set, say V_1 . Then it is possible to find a strong mixed $n(G_M^o)$ -coloring such that all vertices in V_1 have an even color, and all vertices in V_2 have an odd color.*

Proof: Since G_M^o has no circuit, we may decompose its set of vertices into subsets $C_0, C_1, \dots, C_{n(G_M^o)-1}$, where C_i is the set of vertices having no predecessors when vertices in C_0, C_1, \dots, C_{i-1} have been removed.

So we start with the vertices in C_0 and give each vertex v color 0 if it is in V_1 or color 1 if it is in V_2 , and we continue with the vertices in C_1, C_2, \dots , by giving each vertex the smallest

color which is larger than the color of all its predecessors.

This will give an odd color to vertices in V_2 and an even color to vertices in V_1 (since G_M^o is bipartite a vertex in V_1 (resp. V_2) has all its predecessors in V_2 (resp. V_1)). Clearly we will have $c(v) < c(w)$ for each arc (v, w) . Furthermore no more than $n(G_M^o)$ colors will be used (the longest paths starting in V_2 will have length less than $n(G_M^o) - 1$ and therefore contain colors in $\{1, 2, \dots, n(G_M^o) - 1\}$). ■

Now using this Lemma, we obtain the following result.

Theorem 2.2. *Let $G_M = (V_1 \cup V_2, U, E)$ be a mixed bipartite graph. Assume that all paths of length $n(G_M^o) - 1$ start in the same vertex set, say V_1 . Then it is possible to find a strong mixed $n(G_M^o)$ -coloring such that all vertices in V_1 have an even color, and all vertices in V_2 have an odd color.*

Proof: From Lemma 2.1 we know that the vertices of G_M^o can be colored using at most $n(G_M^o)$ colors and such that all vertices in V_1 have an even color and all vertices in V_2 have an odd color. Notice that whenever there is an edge between two colored vertices v and w , we necessarily have that $c(v) \neq c(w)$, since if one color is even, then the second one is odd. By coloring the remaining uncolored vertices of V_1 with color 0 and the remaining uncolored vertices of V_2 with color 1, we obtain a strong mixed $n(G_M^o)$ -coloring such that all vertices in V_1 have an even color and all vertices in V_2 have an odd color. ■

Theorem 2.3. *Let $G_M = (V_1 \cup V_2, U, E)$ be a complete mixed bipartite graph. Then $\gamma_M(G_M) = n(G_M^o)$ if and only if all paths of length $n(G_M^o) - 1$ start in the same vertex set $V_i, i \in \{1, 2\}$.*

Proof: From Theorem 2.2 we know that if these paths start in the same vertex set, then $\gamma_M(G_M) = n(G_M^o)$. Now suppose that the strong mixed chromatic number is equal to $n(G_M^o)$. Assume there are two paths of length $n(G_M^o) - 1$ having their startvertices not in the same vertex set $V_i, i \in \{1, 2\}$; these vertices are necessarily linked by an edge since the graph is complete. But in this case, a proper strong mixed $n(G_M^o)$ -coloring would not be possible. So we conclude that all paths of length $n(G_M^o) - 1$ start in the same vertex set $V_i, i \in \{1, 2\}$. ■

Theorem 2.4. *Let $G_M = (V, U, E)$ be a mixed graph such that $G(V_o)$ has strong mixed chromatic number $\gamma_M(G(V_o)) \leq n(G_M^o) + 1$. Suppose that we have $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) \leq n(G_M^o)$, where G' is an induced subgraph of G_M containing V_o . Then $\gamma_M(G_M) \leq n(G_M^o) + 1$.*

Proof: Consider $G(V_o)$; it can be colored with at most $n(G_M^o) + 1$ colors. Now assume that the above condition holds. We can remove the vertices of set $V \setminus V_o$ by taking at each step a vertex with minimum degree in the remaining graph (this is the Smallest Last Ordering of

[59]); all these degrees will be at most $n(G_M^o)$ as we will now show. So when reinserting the vertices in the opposite order, it will be possible to color the graph with at most $n(G_M^o) + 1$ colors (for each vertex there will be a color available among the $n(G_M^o) + 1$ colors).

Let us call v_1, v_2, \dots, v_q the vertices of $V \setminus V_o$ in the order in which they are removed, and let us call G_i the partial subgraph of G_M remaining when vertices v_1, \dots, v_{i-1} have been removed; so $G_1 = G_M$. We denote by G' an induced subgraph of G_M containing V_o . We have $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) \geq \max_{1 \leq i \leq q} (\min_{v \in G_i, v \notin V_o} (d_{G_i}(v))) = \max_{1 \leq i \leq q} (d_{G_i}(v_i))$ since in the left hand side all possible induced subgraphs G' of G_M containing V_o are considered, while in the right hand side, only G_1, \dots, G_q are considered.

We also have $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) \leq \max_{1 \leq i \leq q} (d_{G_i}(v_i))$. In fact, let G'' be the induced subgraph for which the maximum on the left is attained. Let v_r be the first vertex of G'' which is removed in the above process. Then $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) = \min_{v \in G'', v \notin V_o} (d_{G''}(v)) \leq d_{G''}(v_r) \leq d_{G_r}(v_r) \leq \max_{1 \leq i \leq q} (d_{G_i}(v_i))$. So the above inequality holds. It follows that $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) = \max_{1 \leq i \leq q} (d_{G_i}(v_i)) \leq n(G_M^o)$. Hence the coloring of G_M is possible with at most $n(G_M^o) + 1$ colors. ■

As already mentioned at the beginning of this section, we know that for a mixed bipartite graph G_M , $\gamma_M(G_M) \leq n(G_M^o) + 1$, and so we obtain the following corollary.

Corollary 2.5. *Let G_M be a mixed graph such that $G(V_o)$ is mixed bipartite and such that $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) \leq n(G_M^o)$, where G' is an induced subgraph of G_M containing V_o . Then $\gamma_M(G_M) \leq n(G_M^o) + 1$.*

Corollary 2.6. *Let G_M be a mixed graph such that each odd cycle C in G_M contains at least one vertex which is not incident to any arc and such that $\max_{G' \subseteq G_M} (\min_{v \in G', v \notin V_o} (d_{G'}(v))) \leq n(G_M^o)$, where G' is an induced subgraph of G_M containing V_o . Then $\gamma_M(G_M) \leq n(G_M^o) + 1$.*

Proof: Consider the mixed graph $G(V_o)$. Since each odd cycle in G_M contains at least one vertex which is not incident to any arc, $G(V_o)$ has no odd cycle and hence is mixed bipartite. We conclude by using Corollary 2.5. ■

2.2.2 Complexity results

In this section we will give some complexity results concerning the strong mixed graph coloring problem. In [41] an open question is the complexity to decide whether the strong mixed chromatic number is $n(G_M^o)$ or $n(G_M^o) + 1$ for mixed bipartite graphs. G. Rote has shown with an elementary construction that this problem is \mathcal{NP} -complete [64]. Here we will strengthen this result by proving that it is \mathcal{NP} -complete even for planar bipartite graphs with maximum degree 4 and such that all vertices incident to at least one arc have maximum degree 2 as well as for bipartite graphs with maximum degree 3.

Theorem 2.7. *$S(G_M, 3)$ is \mathcal{NP} -complete even if G_M is planar bipartite with maximum degree 4 and each vertex incident to an arc has maximum degree 2.*

Proof: We use a reduction from the List Coloring problem (*LiCol*) which is defined as follows.

Instance: An undirected graph $G = (V, E)$ together with sets of feasible colors $L(v)$ for all vertices $v \in V$.

Question: Does there exist a proper vertex coloring of G with colors from $L = \bigcup_{v \in V} L(v)$ such that every vertex v is colored with a feasible color from $L(v)$?

This problem is shown to be \mathcal{NP} -complete even if G is a 3-regular planar bipartite graph and the total number of colors is 3 and each list $L(v)$ contains 2 or 3 colors (see [15]).

Let G be a 3-regular planar bipartite graph. Suppose that each vertex v is given a list $L(v)$ with feasible colors such that $2 \leq |L(v)| \leq 3$, and such that the total number of colors is 3 (colors 0,1 and 2). For each vertex v in G such that $|L(v)| = 2$, introduce new vertices as shown in Figure 2.1 depending on the list $L(v)$. The mixed graph G_M we thereby obtain is planar and bipartite with $\Delta(G_M) = 4$; each vertex incident to an arc has maximum degree 2 and $n(G_M^o) = 3$.

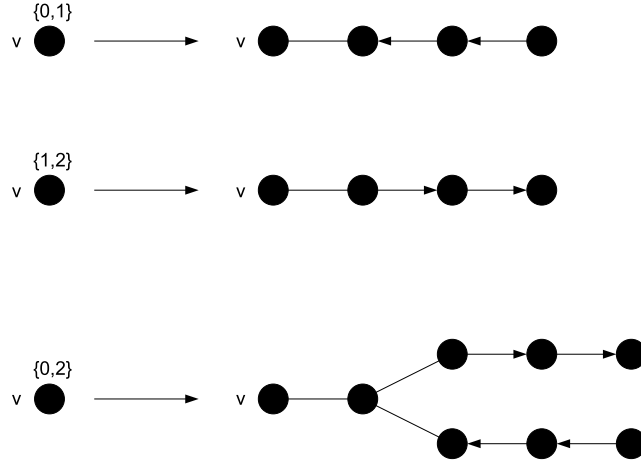


Figure 2.1: Depending on the list $L(v)$, we add new vertices, edges and arcs.

Suppose now that $LiCol(G)$ has a positive answer. Denote by c the coloring corresponding to the solution. Then in G_M , color each vertex v which is also in G with the color $c(v)$. It is easy to see that the remaining uncolored vertices (those which were added) can be colored using colors 0, 1 and 2 such that all the constraints are satisfied.

Conversely, if $S(G_M, 3)$ has a solution, each original vertex gets necessarily a color from its list $L(v)$ in G , and hence we obtain a solution of $LiCol(G)$ in G by removing in G_M the new vertices added at the beginning. ■

Theorem 2.8. $S(G_M, 3)$ is \mathcal{NP} -complete when G_M is bipartite and has maximum degree 3.

Proof: We will use a transformation from the 3SAT problem which is known to be \mathcal{NP} -complete even if each variable appears at most three times, each literal at most twice and each clause contains 2 or 3 literals [61]. Notice that we can assume that whenever a literal appears twice, this literal is positive.

With each variable x , we associate the variable gadget shown in Figure 2.2.

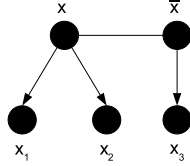


Figure 2.2: The variable gadget.

With each clause $c_e = (x \vee y \vee z)$, where x has its i^{th} occurrence, y its j^{th} occurrence and z its q^{th} occurrence, $i, j, q \in \{1, 2\}$, we associate the variable gadget shown in Figure 2.3(a).

With each clause $c_e = (x \vee y \vee \bar{z})$, where x has its i^{th} occurrence, y its j^{th} occurrence, $i, j \in \{1, 2\}$, we associate the variable gadget shown in Figure 2.3(b).

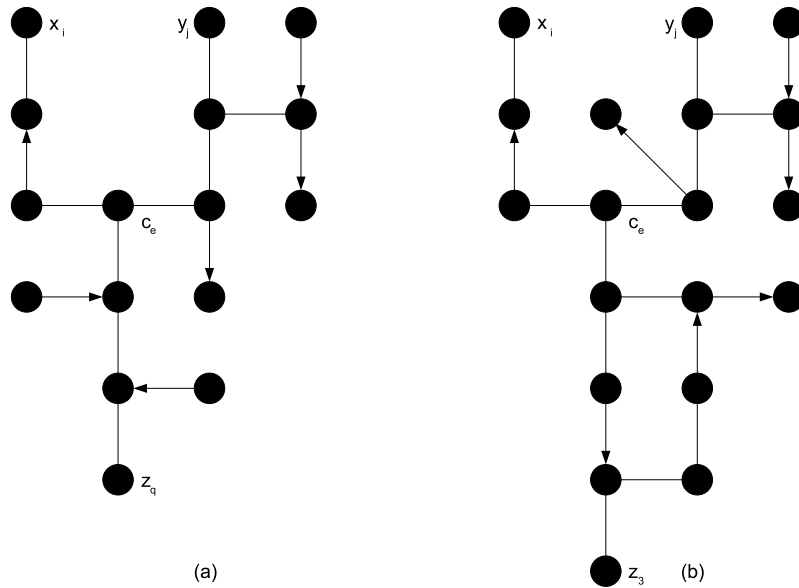


Figure 2.3: (a) The clause gadget for $c_e = (x \vee y \vee z)$. (b) The clause gadget for $c_e = (x \vee y \vee \bar{z})$.

With each clause $c_e = (x \vee \bar{y} \vee \bar{z})$, where x has its i^{th} occurrence, $i \in \{1, 2\}$, we associate the variable gadget shown in Figure 2.4(a).

With each clause $c_e = (\bar{x} \vee \bar{y} \vee \bar{z})$, we associate the variable gadget shown in Figure 2.4(b).

With each clause c_e containing only two literals, say l' and l'' , we associate the clause gadget

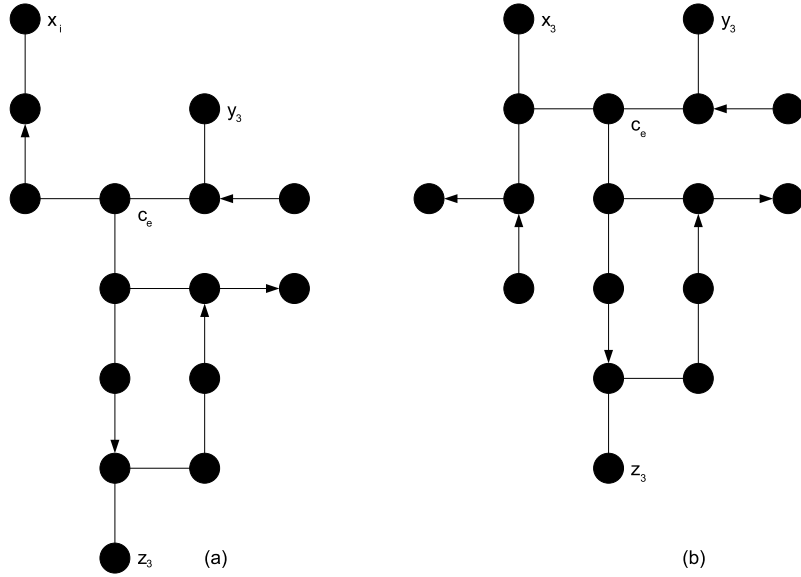


Figure 2.4: (a) The clause gadget for $c_e = (x \vee \bar{y} \vee \bar{z})$. (b) The clause gadget for $c_e = (\bar{x} \vee \bar{y} \vee \bar{z})$.

that we would associate with the clause $(t_e \vee l' \vee l'')$, where t_e is a new variable, but for t_e we introduce the variable gadget shown in Figure 2.5 instead of the variable gadget of Figure 2.2.

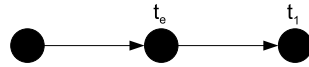


Figure 2.5: Variable gadget for t_e .

The mixed graph we obtain has maximum degree 3. Notice that for every pair of vertices $\{c_e, c_f\}$, a chain from c_e to c_f always has even length. Thus there is no odd cycle in G_M , i.e., G_M is bipartite. Furthermore, the length of a longest directed path is 2.

Also notice that if we want to color the vertices of G_M using only colors 0, 1 and 2, in each clause gadget associated with clause c_e the vertices x_i, y_j, z_q , $i, j, q \in \{1, 2, 3\}$, must not all be colored with color 2, otherwise there would be no more color left for vertex c_e . On the other hand, when at least one of the vertices x_i, y_j, z_q has color 1, the clause gadget can be properly colored with colors 0, 1 and 2.

Suppose there is a truth assignment such that the formula is ‘true’. For each variable x which has value ‘true’, color vertex x with color 0, vertices x_1, x_2, \bar{x} with color 1 and vertex x_3 with color 2. For each variable y which has value ‘false’, color vertex \bar{y} with color 0, vertices y, y_3 with color 1 and vertices y_1, y_2 with color 2. As in each clause there is at least one variable which has value ‘true’, there is in each clause gadget at least one of the vertices x_i, y_j, z_q which has color 1, and thus, as we mentioned before, the clause gadget can be properly colored using colors 0, 1 and 2. So $S(G_M, 3)$ has a positive answer.

Suppose now $S(G_M, 3)$ has a positive answer. In this case, in each variable gadget corre-

sponding to a variable x , one of the vertices x, \bar{x} has color 0 and one has color 1. Set variable x to ‘true’ if x has color 0 and set it to ‘false’ otherwise. As in each clause gadget at least one of the vertices x_i, y_j, z_q has color 1, there is at least one variable in the corresponding clause which will be set to the value ‘true’. Thus we have a truth assignment such that the formula is ‘true’. ■

We will now give some polynomially solvable cases in special classes of graphs. First let us introduce the Precoloring Extension problem (*PrExt*) which is defined as follows.

Instance: An undirected graph $G = (V, E)$ and some vertices of V are precolored properly using at most q colors.

Question: Can this precoloring of G be extended to a proper coloring of G using at most q colors?

This problem was shown to be polynomially solvable in special classes of graphs like split graphs [44], cographs [45], complements of bipartite graphs [44] or graphs of maximum degree 3 [15].

Theorem 2.9. *$S(G_M, n(G_M^o))$ is polynomially solvable if every vertex in G_M^o is on a path of length $n(G_M^o) - 1$ and if the Precoloring Extension problem on the graph G with at most $n(G_M^o)$ colors, obtained by transforming each arc of G_M into an edge, is polynomially solvable.*

Proof: Let G_M be a mixed graph with G_M^o satisfying the above hypothesis and such that $PrExt(G, n(G_M^o))$ is polynomially solvable. Notice that if there exists a strong mixed $n(G_M^o)$ -coloring c of G_M , then each vertex v belonging to G_M^o must get color $c(v) = in(v)$. So we color each vertex v incident to an arc with the color $c(v) = in(v)$. If a conflict occurs, i.e., if there are two adjacent vertices which get the same color, then no solution exists. Otherwise consider all arcs as edges. We get an undirected graph G with some precolored vertices. Thus we get an instance of the Precoloring Extension problem in G . We know that $PrExt(G, n(G_M^o))$ is polynomially solvable. It is easy to see that the two problems are equivalent. Thus our problem is polynomially solvable. ■

By [44] we know that $PrExt(G, 2)$ is polynomially solvable. Thus we deduce the following corollary from Theorem 2.9.

Corollary 2.10. *$S(G_M, 2)$ can be solved in polynomial time.*

Theorem 2.11. *Let G_M be a mixed graph having the following properties:*

- (a) $V = V_o$;
- (b) for all maximal directed paths P in G_M , $|P| = n(G_M^o)$ or $|P| = n(G_M^o) - 1$.

Then deciding whether $\gamma_M(G_M) = n(G_M^o)$ or $\gamma_M(G_M) > n(G_M^o)$ can be done in polynomial time.

Proof: Using an idea based on [68], we transform the problem into a $2SAT$ problem which is known to be polynomially solvable [36]. Denote by \mathcal{P} the set of vertices belonging to a path P with $|P| = n(G_M^o)$.

- (1) with each vertex $v \in \mathcal{P}$ with $in(v) = r$, we associate a variable v_r and a clause (v_r) ;
- (2) with each vertex $v \notin \mathcal{P}$ with $in(v) = r$, we associate two variables v_r and v_{r+1} ;
- (3) with each path $P = (v^0, v^1, \dots, v^{n(G_M^o)-2})$ with $|P| = n(G_M^o) - 1$, we associate the clauses $(v_i^i \vee v_{i+1}^i), (\bar{v}_i^i \vee \bar{v}_{i+1}^i)$, for $i = 0, 1, \dots, n(G_M^o) - 2$, and the clause $(\bar{v}_{j+1}^j \vee \bar{v}_{j+1}^{j+1})$, for $j = 0, 1, \dots, n(G_M^o) - 3$;
- (4) with each edge $[v, w] \in E$ such that $v \in \mathcal{P}$, $w \notin \mathcal{P}$ and $in(v) = in(w) = r$ (resp. $in(v) = in(w) + 1 = r + 1$), we associate the clause $(\bar{v}_r \vee \bar{w}_r)$ (resp. $(\bar{v}_{r+1} \vee \bar{w}_{r+1})$);
- (5) with each edge $[v, w] \in E$ such that $v, w \notin \mathcal{P}$ and $in(v) = in(w) = r$ (resp. $in(v) = in(w) + 1 = r + 1$), we associate the clauses $(\bar{v}_r \vee \bar{w}_r), (\bar{v}_{r+1} \vee \bar{w}_{r+1})$ (resp. $(\bar{v}_{r+1} \vee \bar{w}_{r+1})$);
- (6) with each edge $[v, w] \in E$ such that $v, w \in \mathcal{P}$ and $in(v) = in(w) = r$, we associate the clause $(\bar{v}_r \vee \bar{w}_r)$.

Suppose that an instance of $2SAT$ is ‘true’. If a variable v_r is set to ‘true’, then we will color the corresponding vertex v with color r , i.e. $c(v) = r$. Notice that each vertex $v \in \mathcal{P}$ will be colored with $c(v) = in(v)$ (see (1)), and each vertex $v \notin \mathcal{P}$ will be colored with $c(v) = in(v)$ or $c(v) = in(v) + 1$ (see (2) and (3)). Thus the coloring uses at most $n(G_M^o)$ colors. The clauses in (1) and (3) ensure that for all $(v, u) \in U$ we have $c(v) < c(u)$, and the clauses in (1), (4), (5) and (6) ensure that for all $[v, w] \in E$, $c(v) \neq c(w)$. So we conclude that $\gamma_M(G_M) = n(G_M^o)$.

Suppose now that $\gamma_M(G_M) = n(G_M^o)$. Notice that in that case each vertex $v \in \mathcal{P}$ will be colored with $c(v) = in(v)$, and each vertex $v \notin \mathcal{P}$ will be colored with $c(v) = in(v)$ or $c(v) = in(v) + 1$. For each variable v_r occurring in the formula, set it to ‘true’ if $c(v) = r$ and ‘false’ otherwise. It is easy to see that each clause will be satisfied. ■

The previous theorem has the following consequence.

Corollary 2.12. *Let G_M be a mixed graph having the following properties:*

- (a) $V = V_o$;
- (b) $n(G_M^o) \leq \gamma_M(G_M) \leq n(G_M^o) + 1$;
- (c) for all maximal directed paths P in G_M , $|P| = n(G_M^o)$ or $|P| = n(G_M^o) - 1$.

Then the strong mixed chromatic number of G_M can be determined in polynomial time.

We denote by n_i the number of vertices on a longest directed path P in G_M containing vertex v_i (if v_i is not incident to any arc, $n_i = 1$ and $P = \{v_i\}$). Notice that $n_i = in(v_i) + out(v_i) + 1$. Let $h \geq |P|$ be an integer. We define S_i as the set of possible colors for v_i such that whenever v_i has a color $c(v_i) \in S_i$ there exists a coloring c of G_M (with an arbitrary number of colors) with $c(v) \leq h - 1$, for any $v \in P$. We have the following result.

Proposition 2.13. *Let $P = \{v_1, v_2, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_{n_i}\}$ be a longest directed path in G_M containing v_i and let $h \geq |P|$ be an integer. Then $S_i = \{in(v_i), in(v_i) + 1, \dots, h - (out(v_i) + 1)\}$.*

Proof: It is easy to see that the smallest feasible color for v_i is $in(v_i)$. Suppose that $c(v_i) = in(v_i) + q$, where $q \geq 0$. We can color the vertices v_1, v_2, \dots, v_{i-1} with colors $c(v_1) = in(v_1), c(v_2) = in(v_2), \dots, c(v_{i-1}) = in(v_{i-1})$ and vertices v_{i+1}, \dots, v_{n_i} with colors $c(v_{i+1}) = in(v_i) + q + 1, \dots, c(v_{n_i}) = in(v_i) + q + n_i - i$. Notice that $n_i - i = out(v_i)$ since P is a longest directed path containing v_i . Thus $c(v_{n_i}) = in(v_i) + q + out(v_i)$. This way we get a feasible coloring c of G_M (the vertices of G_M not belonging to P can easily be properly colored) and since the condition $c(v) \leq h - 1$ must hold for any $v \in P$, we have that $in(v_i) + q + out(v_i) \leq h - 1$, i.e., $q \leq h - (in(v_i) + out(v_i) + 1)$. Thus $S_i = \{in(v_i), in(v_i) + 1, \dots, h - (out(v_i) + 1)\}$. ■

We will now focus on a special class of graphs: partial p -trees. A p -tree is a graph defined recursively as follows. A p -tree on p vertices consists of a p -clique. Given any p -tree T_n on n vertices, we construct a p -tree on $n + 1$ vertices by adjoining a new vertex v_{n+1} to T_n , which is made adjacent to each vertex of some p -clique of T_n and nonadjacent to the remaining $n - p$ vertices. A *partial p -tree* is a partial subgraph of a p -tree. Here we shall consider partial p -trees for $p \geq 2$.

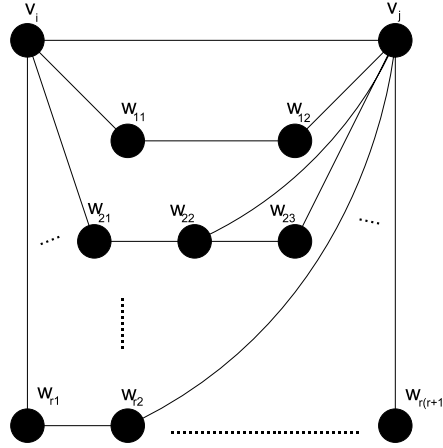
Consider now an undirected partial p -tree $G = (V, E)$. Suppose that for some edges $[v_i, v_j] \in E$ we add new vertices and edges as shown in Figure 2.6. Denote by G' the graph we obtain.

We have the following result.

Proposition 2.14. *Let G be a partial p -tree. Then G' is also a partial p -tree.*

Proof: Since G is a partial p -tree, it is the partial subgraph of a p -tree T_p . Notice that $[v_i, v_j] \in K_{ij}$ in T_p , where K_{ij} is a $(p + 1)$ -clique. Consider T' which is the graph obtained by adding to G' all the edges and vertices of T_p which are not in G . In order to show that G' is a partial p -tree, we just need to show how edges can be added to T' to make it become a p -tree T^* .

For each new vertex w_{s1} , $s = 1, \dots, r$, make it adjacent to v_j and to $p - 2$ vertices in $K_{ij} - \{v_i, v_j\}$. We obtain, for each s , a $(p + 1)$ -clique K_{s1} containing w_{s1} . Each new vertex w_{st} , $s = 1, \dots, r$, $t = 2, \dots, r + 1$, is linked to $p - 2$ vertices in $K_{s(t-1)} - \{w_{s(t-1)}, v_j\}$. We obtain then for each s and t , $t \neq 1$, a $(p + 1)$ -clique K_{st} containing w_{st} . Clearly the resulting graph is a p -tree and thus G' is a partial p -tree. ■


 Figure 2.6: New vertices and edges added to $[v_i, v_j]$.

In [33], it is shown that $S(G_M, k)$ is polynomially solvable for series parallel graphs, i.e., partial 2-trees, by giving an algorithm which has complexity $O(n^{3.376} \log(n))$. Here we will give a result concerning general partial p -trees, for fixed p .

Theorem 2.15. *$S(G_M, k)$ is polynomially solvable if $G_M = (V, U, E)$ is a partial p -tree for fixed p .*

Proof: We use a transformation to the *LiCol* problem which is known to be solvable in $O(n^{p+2})$ time for partial p -trees (see [45]).

For each $v_l \in V$ which is not incident to any arc, we set $L(v_l) = \{0, 1, \dots, k-1\}$. For each vertex $v_i \in V$ which is incident to at least one arc, we set $L(v_i) = \{in(v_i), in(v_i) + 1, \dots, k - (out(v_i) + 1)\}$. For each arc $(v_i, v_j) \in U$ such that $k - (out(v_i) + 1) > in(v_j)$, we introduce new vertices and edges as shown in Figure 2.6 with $r = k - (out(v_i) + in(v_j) + 1)$. For the new vertices we set:

$$L(w_{st}) = \begin{cases} \{in(v_j) + s, in(v_j) + s + 1\} & \text{if } 1 \leq s \leq r \text{ and } t = 1; \\ \{in(v_j) + s + 1, in(v_j)\} & \text{if } 1 \leq s \leq r \text{ and } t = 2; \\ \{in(v_j) + t - 3, in(v_j) + t - 2\} & \text{if } 1 \leq s \leq r \text{ and } 3 \leq t \leq r + 1. \end{cases}$$

Figure 2.7 shows a case where we have $L(v_i) = \{3, 4, 5, 6\}$, $L(v_j) = \{4, 5, 6, 7\}$ and $k = 8$. For the new vertices we set $L(w_{11}) = \{5, 6\}$, $L(w_{12}) = \{6, 4\}$, $L(w_{21}) = \{6, 7\}$, $L(w_{22}) = \{7, 4\}$ and $L(w_{23}) = \{4, 5\}$. This way we do not allow vertex v_j to get a color less than the color of vertex v_i .

By considering all arcs as edges, we obtain a new undirected graph G' which is still a partial p -tree (see Proposition 2.14). Furthermore we associated with each vertex v in G' a list $L(v)$ of integers such that $L(v) \subseteq \{0, 1, \dots, k-1\}$. Thus, we get an instance of the *LiCol* problem with k colors in a partial p -tree G' , where p is fixed.

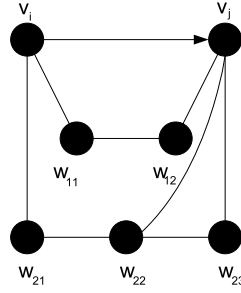


Figure 2.7: Example of how new edges and vertices are introduced in the case of $L(v_i) = \{3, 4, 5, 6\}$ and $L(v_j) = \{4, 5, 6, 7\}$.

Suppose that an instance of the $LiCol(G')$ problem has answer ‘yes’ and denote by c the corresponding list-coloring. We will show that c restricted to G_M is also a feasible coloring for $S(G_M, k)$. For each edge $[v_i, v_j]$ in G' , we have that $c(v_i) \neq c(v_j)$ and so $c(v_i) \neq c(v_j)$ for each $[v_i, v_j]$ or (v_i, v_j) in G_M . Consider now an arc (v_i, v_j) in G_M . We have to verify that $c(v_i) < c(v_j)$. If $c(v_i) \leq in(v_j)$, we clearly have that $c(v_i) < c(v_j)$. So suppose that $c(v_i) = in(v_j) + q$, $q > 0$. In that case vertex w_{q1} has necessarily color $in(v_j) + q + 1$ and vertices $w_{q2}, \dots, w_{q(q+1)}$ must have colors $in(v_j), \dots, in(v_j) + q - 1$, due to their lists. Since these vertices are adjacent to v_j , $c(v_j) > in(v_j) + q$ and hence $c(v_i) < c(v_j)$. We conclude that $S(G_M, k)$ has answer ‘yes’.

Conversely, suppose now that an instance of $S(G_M, k)$ has answer ‘yes’, and denote by c' the corresponding strong mixed k -coloring. Then each vertex v in G_M has a color which belongs to the corresponding list $L(v)$ in G' , i.e., $c'(v) \in L(v)$. In fact, for each vertex v_l not adjacent to any arc in G_M , we have $L(v_l) = \{0, 1, \dots, k - 1\}$ and for each vertex v_i which is incident to at least one arc, we have $L(v_i) = \{in(v_i), in(v_i) + 1, \dots, k - (out(v_i) + 1)\}$. By Proposition 2.13, we know that these colors are the only possible if P_{v_i} (a longest directed path containing v_i) is properly colored and $c'(v) < k$, for any $v \in P_{v_i}$. Furthermore it is not difficult to verify that coloring c' can easily be extended in G' by coloring the new vertices w_{st} (using the colors in their associated lists) and so we get a feasible coloring for the $LiCol$ problem in G' . Thus the $LiCol$ problem on G' has answer ‘yes’.

Clearly G' can be obtained from G_M in polynomial time since n_i and $in(v_i)$ can be computed in polynomial time for each vertex v_i in G_M . The number of new vertices is restricted by $O(n^2m)$, where m is the number of arcs, and thus $S(G_M, k)$ can be solved in time $O(n^{2p+4}m^{p+2})$ if G_M is a partial p -tree, with fixed p . ■

2.3 Weak mixed graph coloring problem

In this section we study the following problem which we call the *weak mixed graph coloring problem*.

Instance: A mixed graph $G_M = (V, U, E)$ and an integer $k \geq 1$.

Question: Can the vertices of G_M be colored using at most k colors such that for each edge $[v_i, v_j] \in E$, $c(v_i) \neq c(v_j)$ and for each arc $(v_l, v_q) \in U$, $c(v_l) \leq c(v_q)$?

We will refer to this problem as $W(G_M, k)$. Notice that we have $\chi_M(G_M) \leq \gamma_M(G_M)$.

Necessary and sufficient conditions for a mixed graph to admit a weak mixed coloring have been given.

Theorem 2.16 (see for instance [69, 70]). *For the existence of a weak mixed coloring of a mixed graph $G_M = (V, U, E)$ it is necessary and sufficient that graph (V, \emptyset, E) does not have loops and that G_M does not contain any directed circuit with a chord.*

In the rest of this section, we will suppose that these conditions are satisfied. Notice that in the case of weak mixed coloring we may have $(v_l, v_q) \in U$ and $[v_l, v_q] \in E$. Then in any proper weak mixed coloring c , we must have $c(v_l) < c(v_q)$. So the strong mixed graph coloring problem $S(G_M, k)$ is the special case of $W(G_M, k)$, where for each arc $(v_l, v_q) \in U$ we have $[v_l, v_q] \in E$.

2.3.1 Bounds on the weak mixed chromatic number

We will start with a few observations which will allow us to simplify the original mixed graph G_M (see also [72] where a similar merging operation is designed for vertices belonging to the same strongly connected component).

Lemma 2.17. *Let $G_M = (V, U, E)$ be a mixed graph and let C be a strongly connected component of G_M^o . Then, in any feasible weak mixed coloring c of G_M , $c(v_i) = c(v_j) \forall v_i, v_j \in C$.*

Proof: Let c be a feasible coloring of G_M . Suppose that $c(v_i) < c(v_j)$ for some v_i, v_j in C . Since there is a directed path from v_j to v_i contained in C , we obtain a contradiction because we should have $c(v_j) \leq c(v_i)$. ■

Consider a mixed graph G_M and let $\mathcal{D} = \{D_1, \dots, D_t\}$ be a set of disjoint directed partial subgraphs of G_M . Let G_M/\mathcal{D} be the mixed graph obtained by deleting the arcs of $\bigcup_{l=1}^t D_l$ and by replacing the vertices of each graph D_l by a single vertex v_l . G_M/\mathcal{D} may have multiple edges or arcs in which case we delete them. We say that D_l has been *contracted* to a single vertex v_l , for all $l = 1, \dots, t$. Then we have the following result.

Lemma 2.18. *Let $G_M = (V, U, E)$ be a mixed graph and let $\mathcal{C} = \{C_1, C_2, \dots, C_q\}$ be the set of strongly connected components in G_M^o such that $\forall v, w \in C_i, i \in \{1, 2, \dots, q\}$, we have $[v, w] \notin E$. Then $\chi_M(G_M) = \chi_M(G_M/\mathcal{C})$.*

Proof: Let c be an optimal coloring of G_M/\mathcal{C} . Let v_i be the vertex in G_M/\mathcal{C} representing component C_i and let $c(v_i)$ be its color, for $i = 1, 2, \dots, q$. Consider now G_M and color each vertex $w \notin C_i, i = 1, 2, \dots, q$, in G_M with the same color as in G_M/\mathcal{C} . Color each vertex in C_i with color $c(v_i)$. Clearly we obtain a feasible coloring of G_M . Furthermore this coloring is optimal. In fact, suppose that G_M can be colored with $\chi_M(G_M) < \chi_M(G_M/\mathcal{C})$ colors. By Lemma 2.17, we know that all vertices of $C_i, i = 1, 2, \dots, q$, have necessarily the same color c_i . Contracting each component C_i to a single vertex v_i and coloring it with color c_i , we obtain a feasible coloring of G_M/\mathcal{C} with $\chi_M(G_M) < \chi_M(G_M/\mathcal{C})$ colors, which is a contradiction. ■

Consider a mixed graph $G_M = (V, U, E)$. As we have seen in Lemma 2.18, all strongly connected components of G_M^o such that no two vertices of a same component are linked by an edge can be contracted to single vertices without changing the weak mixed chromatic number of the original graph. So from now on we suppose that in G_M all these strongly connected components have been contracted to single vertices. Let v be a vertex of G_M which is not incident to any edge. Denote by $Pred(v)$ the set of its neighbors w such that $(w, v) \in U$ and by $Succ(v)$ the set of its neighbors u such that $(v, u) \in U$. Delete vertex v from G_M and introduce arcs (w, u) for all $w \in Pred(v)$ and $u \in Succ(v)$. Suppose we perform this operation as long as there is a vertex v which is not incident to any edge. Let $G_M^* = (V^*, U^*, E)$ be the mixed graph obtained. Then we have the following result.

Theorem 2.19. *Let $G_M = (V, U, E)$ be a mixed graph. Then $\chi_M(G_M) = \chi_M(G_M^*)$.*

Proof: Consider an optimal weak mixed coloring of G_M^* . This coloring c can be extended to an optimal weak mixed coloring of G_M . In fact, consider the mixed graph G_M and color each vertex v which is incident to at least one edge with color $c(v)$. Now color each remaining uncolored vertex v (incident to no edge) with color $c(v) = \max_{w \in Pred(v)}(c(w))$. We obtain a feasible weak mixed coloring of G_M . Furthermore this coloring is optimal. Suppose that it is possible to color G_M with k colors, $k < \chi(G_M^*)$. Then by transforming G_M into G_M^* we obtain a feasible coloring of G_M^* with at most k colors, which is a contradiction. Thus $\chi_M(G_M) = \chi_M(G_M^*)$. ■

So from now on we can also suppose that G_M does not contain any vertex incident only to arcs.

Let us consider the set \mathcal{DP} of all maximal directed paths in G_M . Let $P = (v_1, \dots, v_r)$ be a maximal directed path and $E_P = \{[v_i, v_j] | 0 < i < j \leq r\}$ the set of edges linking each a pair of vertices of P . We denote by E_P^1, \dots, E_P^t the subsets of E_P such that if $[v_i, v_j], [v_k, v_l] \in E_P^s$,

then $\max(i, j) \leq \min(k, l)$, for $s = 1, \dots, t$. If $e_P = \max_{s=1, \dots, t}(|E_P^s|)$, then we obtain the following lower bound on the weak mixed chromatic number.

Theorem 2.20. *Let G_M be a mixed graph. Then $\max_{P \in \mathcal{DP}}(e_P + 1) \leq \chi_M(G_M)$.*

Proof: Let $P' = (v'_1, \dots, v'_q)$ be a maximal directed path such that $P' = \arg \max_{P \in \mathcal{DP}}(e_P + 1)$. Suppose that $e_{P'} = |E_{P'}^f|$ for a certain integer f and $E_{P'}^f = \{[v'_{i_1}, v'_{i_2}], [v'_{i_3}, v'_{i_4}], \dots, [v'_{i_{r-1}}, v'_{i_r}]\}$, $0 < i_1 < i_2 \leq i_3 < \dots \leq i_{r-1} < i_r \leq q$.

If we want to construct a weak mixed graph coloring c , we must have $c(v'_{i_j}) < c(v'_{i_{j+1}})$, for $j = 1, 3, \dots, r - 1$ since there is a directed path from v'_{i_j} to $v'_{i_{j+1}}$ and there is an edge $[v'_{i_j}, v'_{i_{j+1}}]$ for all $j = 1, 3, \dots, r - 1$. Furthermore we can color each vertex v'_{i_k} with the same color as $v'_{i_{k-1}}$ for $k = 3, 5, \dots, r - 1$. In fact there cannot be any edge between two vertices v'_{i_h}, v'_{i_g} , $i_{k-1} \leq i_h < i_g \leq i_k$ for $k = 3, 5, \dots, r - 1$ otherwise $|E_{P'}^f|$ would not be maximal. Thus we use at least $e_{P'} + 1$ colors. ■

Remark 2.21. *The lower bound given in Theorem 2.20 is tight. Indeed, if for all edges $[v_i, v_j] \in E$, we have $(v_i, v_j) \in U$ or $(v_j, v_i) \in U$, then $\max_{P \in \mathcal{DP}}(e_P + 1) = \chi_M(G_M)$.*

We will give now two very simple classes of graphs for which we can determine the exact value of the weak mixed chromatic number.

Theorem 2.22. *Let $T_M = (V, U, E)$ be a mixed tree, $E \neq \emptyset$. Then $\chi_M(T_M) = 2$.*

Proof: Choose a root r in T_M . Color it with color $c(r) \in \{0, 1\}$. As long as there is an uncolored vertex, choose such a vertex v having one colored neighbor w (it is easy to see that this is always possible). If $[v, w] \in E$, color v with color $c(v) = 1 - c(w)$ and if (v, w) or $(w, v) \in U$, color it with $c(v) = c(w)$.

We will only use two colors and $\forall [v, w] \in E$, $c(v) \neq c(w)$ and $\forall (v, w) \in U$, $c(v) = c(w)$ and hence the conditions are satisfied. We conclude that $\chi_M(T_M) = 2$. ■

Theorem 2.23. *Let $C_M = (V, U, E)$ be a mixed chordless cycle with $U, E \neq \emptyset$. Then $\chi_M(C_M) = 2$.*

Proof: We distinguish two cases:

- (1) if $|E|$ is even;

We contract each arc (v, w) to a single vertex vw . We get an undirected even cycle which we can color with 2 colors. A feasible 2-coloring of C_M is obtained by expanding each vertex vw and by coloring the vertices of the corresponding arc with the same color as vertex vw .

- (2) if $|E|$ is odd;

We choose an arc (v, w) . Contract all arcs (v', w') to single vertices $v'w'$ except arc (v, w) . We get an even cycle containing a single arc (v, w) which we can properly color

using exactly two colors. A feasible 2-coloring of C_M is obtained by expanding each vertex $v'w'$, and by coloring the vertices of the corresponding arc with the same color as vertex $v'w'$.

■

2.3.2 Complexity results

In this section we will give some complexity results concerning the weak mixed graph coloring problem for some special classes of graphs.

Theorem 2.24. *$W(G_M, 3)$ is \mathcal{NP} -complete even if G_M is planar bipartite with maximum degree 4.*

Proof: We use a reduction from $S(G_M, 3)$ which we have shown to be \mathcal{NP} -complete even if G_M is planar bipartite with maximum degree 4 and each vertex incident to an arc has maximum degree 2. Let G_M be such a mixed graph. We replace each arc (v, w) by a path (v, u, z, w) , where u and z are new vertices, and we introduce an edge $[v, w]$. The mixed graph G'_M obtained is planar bipartite and has maximum degree 4.

Suppose that $S(G_M, 3)$ has a positive answer. Then by keeping this coloring c in G'_M and by coloring the new vertices u, z with color $c(v)$, we obtain a solution for our problem. Conversely if $W(G_M, 3)$ has a positive answer, then we color in G_M each vertex v with the same color it gets in G'_M . Clearly we obtain a solution for $S(G_M, 3)$. ■

Remark 2.25. *Notice that in the mixed graph G'_M , vertices which are incident to an arc may have a degree greater than two.*

If we consider a mixed graph G_M such as constructed in the proof of Theorem 2.7, then our problem $W(G_M, k)$ is trivial: we can color G_M using only two colors. In fact, the initial undirected planar cubic bipartite graph G is 2-colorable and it is easy to see that the added vertices can be properly colored (with respect to the weak mixed graph coloring problem) using the same two colors. Hence for this particular class of planar bipartite graphs, $S(G_M, 3)$ is \mathcal{NP} -complete while $W(G_M, k)$ is trivial, for any $k > 1$.

Theorem 2.26. *$W(G_M, 3)$ is \mathcal{NP} -complete even if G_M is bipartite with maximum degree 3.*

Proof: We use a reduction from $S(G_M, 3)$ which we have shown to be \mathcal{NP} -complete if G_M is bipartite with maximum degree 3. In G_M , replace each arc (v, w) by a directed path $(v, u_1, u_2, u_3, u_4, w)$ and add a new edge $[u_1, u_4]$. The resulting graph G'_M is bipartite with maximum degree 3.

Suppose that $W(G'_M, 3)$ has a positive answer. Denote by c the coloring. Then for each

pair of vertices v, w such that $(v, w) \in G_M$, we must have $c(v) < c(w)$ because of the edge $[u_1, u_4]$. Thus by replacing again the directed path by the arc (v, w) and by keeping the coloring c for the vertices of G_M , we obtain a solution for $S(G_M, 3)$. Similarly, if $S(G_M, 3)$ has a positive answer, denote by c' the coloring. Consider the mixed graph G'_M and keep the coloring c' for the vertices of G'_M which are also vertices of G_M . By coloring the new vertices u_1, u_2 and u_3 with color $c'(v)$ and vertex u_4 with color $c'(w)$, we clearly obtain a solution for $W(G'_M, 3)$. ■

Theorem 2.27. $W(G_M, 2)$ is polynomially solvable.

Proof: We shall transform our problem into a $2SAT$ problem which is known to be polynomially solvable (see [36]). Consider a mixed graph G_M . For each vertex x in G_M , we introduce two variables x_0 and x_1 as well as two clauses $(x_0 \vee x_1)$ and $(\bar{x}_0 \vee \bar{x}_1)$. For each edge $[x, y] \in E$, we introduce two clauses $(\bar{x}_0 \vee \bar{y}_0)$ and $(\bar{x}_1 \vee \bar{y}_1)$. Finally, for each arc $(x, y) \in U$, we introduce a clause $(\bar{x}_1 \vee \bar{y}_0)$. Thus we get an instance of $2SAT$.

Suppose that the $2SAT$ instance is ‘true’. Then by coloring each vertex x with color 0 if x_0 is ‘true’, and with color 1 if x_1 is ‘true’, we get a feasible 2-coloring of G_M . Conversely if G_M admits a feasible 2-coloring, then by setting variable x_i to ‘true’ if x has color i , $i \in \{0, 1\}$, we get a positive answer for the $2SAT$ instance. ■

Theorem 2.28. $W(G_M, k)$ is polynomially solvable if G_M is a partial p -tree, for fixed p .

Proof: We will use a similar proof as for the case of strong mixed graph coloring in partial p -trees. Let $G_M = (V, U, E)$ be a mixed partial p -tree, for some fixed p . With each vertex $v \in V$ we associate a list $L(v) = \{0, 1, \dots, k-1\}$ of possible colors. Notice that each list contains all possible colors $0, 1, \dots, k-1$. Now for each arc $(v_i, v_j) \in U$, we introduce new vertices and edges as shown in Figure 2.6 with $r = k-1$. For these new vertices we set:

$$L(w_{st}) = \begin{cases} \{s, s+1\} & \text{if } 1 \leq s \leq r \text{ and } t = 1, \\ \{s+1, 0\} & \text{if } 1 \leq s \leq r \text{ and } t = 2, \\ \{t-3, t-2\} & \text{if } 1 \leq s \leq r \text{ and } 3 \leq t \leq r+1. \end{cases}$$

Remember that the graph we obtain is also a partial p -tree for the same fixed p (see Proposition 2.14). By deleting the arcs, we still have a partial p -tree. So consider the partial p -tree G' obtained by deleting the arcs. Since with each vertex in G' we have associated a list of possible colors, we get an instance of the $LiCol$ problem, which is polynomially solvable in partial p -trees, for fixed p [45]. By using similar arguments as in Theorem 2.15, one can easily prove that $W(G_M, k)$ and $LiCol(G')$ are equivalent and thus $W(G_M, k)$ is polynomially solvable. ■

2.4 Conclusion

We considered two coloring problems in mixed graphs. In the first one, we were interested in coloring the vertices of the graph such that two vertices linked by an edge get different colors and the tail of an arc must get a color which is strictly smaller than the color of the head of the arc. We gave some bounds on the minimum number of colors necessary to color the vertices of special classes of graphs as well as some complexity results. In particular we showed that the strong mixed graph coloring problem is \mathcal{NP} -complete even if the mixed graph is planar bipartite with maximum degree 4 and each vertex incident to an arc has maximum degree 2 or if the graph is bipartite with maximum degree 3. Furthermore we proved that the problem is polynomially solvable in partial p -trees, for fixed p .

In the second problem, we were interested in coloring the vertices of the graph such that two vertices linked by an edge get different colors and the tail of an arc must not get a color larger than the head of the arc. Again we gave some bounds on the minimum number of colors necessary to color the vertices and some complexity results. In particular we showed that this problem is polynomially solvable in partial p -trees, for fixed p .

The results presented here concerned special classes of graphs. Further research is needed to extend these results to other classes of graphs. In particular it would be interesting to know the complexity of the two problems in planar cubic bipartite graphs. Also mixed graphs with a particular structure of the directed graph G_M^o should be analyzed to detect maybe more polynomially solvable cases.

Chapter 3

Bicolored matchings in some classes of graphs

Introduction

Various types of packing problems in graphs have been extensively studied by many authors; the maximum stable set problem (find a maximum cardinality set of mutually non adjacent vertices), the maximum matching problem (find a maximum cardinality set of mutually non adjacent edges) and the maximal forest problem are some of the most famous examples (see for instance [9] for a formulation of many basic packing problems in graph theory).

A natural extension of packing problems has been considered in several forms. It consists in giving a bicoloring (R, B) (for red and black) of the vertex set (resp. of the edge set) of a graph $G = (V, E)$; it is then required to find if G contains a stable set S (resp. a matching M) such that $|S \cap R| \geq p$ and $|S \cap B| \geq q$ (resp. $|M \cap R| \geq p, |M \cap B| \geq q$) where p and q are given positive integers (see [34, 75]).

Besides this, the problem of constructing a spanning tree T in a graph $G = (V, E)$ whose edge set is partitioned into sets R, B is considered in [34] with the requirement that $|T \cap R| \geq p$: it is shown that a solution can easily be constructed by using simple adaptations of basic algorithms.

In addition, the problem of constructing a bicolored perfect matching M in a complete bipartite graph $K_{n,n}$ (both the left set and the right set consist of exactly n vertices) is considered in [75] with the requirement that $|M \cap R| = p$; it is a special case of the problem consisting of determining whether in a complete bipartite graph $K_{n,n}$ where each edge $[i, j]$ has a weight w_{ij} there exists a perfect matching M with weight $w(M) \equiv \sum_{[i,j] \in M} w_{ij} = p$. This general case was shown to be \mathcal{NP} -complete in [13], while the special case $w_{ij} = 1$ if $[i, j] \in R$, and $w_{ij} = 0$ else, was solved with a polynomial algorithm in [49, 75]. The complexity of the problem with 0,1-weights in general bipartite graphs is apparently unknown.

In this chapter we intend to consider a related problem which is also based on a bicoloring

(R, B) of the edge set of a graph. We will essentially try to characterize minimum sets R for which maximum matchings M can be found with specific values of $p = |M \cap R|$.

More specifically, given a graph $G = (V, E)$ and a set $\mathcal{P} = \{p_0, p_1, \dots, p_s\}$ of integers, $0 \leq p_0 < p_1 < \dots < p_s \leq \lfloor |V|/2 \rfloor$, we want to color a subset $R \subseteq E$ of edges of G , say in red, in such a way that for any i ($0 \leq i \leq s$), G contains a maximum matching M_i with exactly p_i red edges, i.e., $|M_i \cap R| = p_i$.

We shall in particular be interested in finding a smallest subset R for which the required maximum matchings do exist.

A subset R will be \mathcal{P} -feasible for G if for every p_i in \mathcal{P} there is a maximum matching M_i in G with $|M_i \cap R| = p_i$. Notice that for some \mathcal{P} there may be no \mathcal{P} -feasible set R (take $\mathcal{P} = \{0, 1, 2\}$ in $G = K_{2,2}$).

In Section 3.1 we will derive some elementary properties of solutions in regular bipartite graphs. Section 3.2 will be devoted to a special case where the set of values $p = |M \cap R|$ is an interval of consecutive integers. Finally Section 3.3 will contain conclusions and possible extensions.

Basic properties of matchings are to be found in [56]. For definitions linked to complexity, the reader is referred to Chapter 1 and [36]. In general all graphs will be simple (i.e., no multiple edges, no loops).

3.1 Regular bipartite graphs

In this section we will state some basic results concerning \mathcal{P} -feasible sets in regular bipartite graphs.

Proposition 3.1. *In a Δ -regular bipartite graph G , for any \mathcal{P} with $|\mathcal{P}| \leq \Delta$, there exists a \mathcal{P} -feasible set R .*

This follows from the fact that the edge set of G can be partitioned into Δ perfect (and hence maximum) matchings (by the König theorem, see Chapter 1).

Let us now briefly consider a special case for a Δ -regular bipartite graph.

Theorem 3.2. *Let $G = (X, Y, E)$ with $|X| = |Y| = n$ be a Δ -regular bipartite graph, with $\Delta \geq 2$, and let $\mathcal{P} = \{p, q\}$, with $1 \leq p < q \leq n$. The minimum cardinality of a \mathcal{P} -feasible set R is given by*

$$|R| = q + \max\{0, p - n + |\mathcal{C}|/2\}$$

where \mathcal{C} is a collection of vertex disjoint cycles which are alternating with respect to a perfect matching, and which have a minimum total length $|\mathcal{C}| = \sum_{C_i \in \mathcal{C}} |C_i|$ satisfying $|\mathcal{C}|/2 \geq q - p$.

Proof: Observe first that one can always find a collection \mathcal{C} of alternating cycles satisfying $|\mathcal{C}|/2 \geq q - p$. Take any 2-factor F in G . It is a collection of vertex disjoint cycles which are alternating with respect to a perfect matching. F exists since G is bipartite and Δ -regular.

Since $0 < q - p \leq n = |F|/2$ the 2-factor F satisfies the inequality. In order to minimize the size of R , we will have to find a family \mathcal{C} which in addition has a minimum number of edges. Let now M be a perfect matching in G .

(1) Assume first $p \leq n - |\mathcal{C}|/2$.

Then we color in red p edges of $M - \mathcal{C}$ and $q - p$ edges of $M \cap \mathcal{C}$. This is possible since $0 < q - p \leq |\mathcal{C}|/2$. Clearly we will have $|M \cap R| = q$; by interchanging the edges of $M \cap \mathcal{C}$ and of $\mathcal{C} - M$ we get a perfect matching M' with $|M' \cap R| = p$. In such a case $|R|$ has minimum cardinality, since we must have $|R| \geq q$ for any \mathcal{P} -feasible set R . Notice that in this case any \mathcal{C} with $|\mathcal{C}|/2 \geq q - p$ will do.

(2) Suppose now $p > n - |\mathcal{C}|/2$.

We color in red $n - |\mathcal{C}|/2$ edges of $M - \mathcal{C}$ and we also color $q - (n - |\mathcal{C}|/2) = q - n + |\mathcal{C}|/2$ edges of $M \cap \mathcal{C}$ as well as $p - (n - |\mathcal{C}|/2) = p - n + |\mathcal{C}|/2$ edges of $\mathcal{C} - M$. This is possible since $0 < p - n + |\mathcal{C}|/2 < q - n + |\mathcal{C}|/2 \leq |\mathcal{C}|/2$. So we have $|R| = p + q - n + |\mathcal{C}|/2 \leq n + |\mathcal{C}|/2$. Again $|M \cap R| = q$ and by interchanging the edges of $M \cap \mathcal{C}$ and of $\mathcal{C} - M$ we get a perfect matching M' with $|M' \cap R| = p$.

In order to have a matching M and a matching M' having respectively q and $p < q$ red edges, M' must be obtained from M by using a collection \mathcal{C} of vertex disjoint alternating cycles with $|\mathcal{C}|/2 \geq q - p$, since $|\mathcal{C}| = |M \Delta M'|$; we would otherwise have $0 < |M \cap R| - |M' \cap R| \leq |\mathcal{C}|/2 < q - p$. Now for any \mathcal{P} -feasible set R which is minimal (inclusionwise), we have two perfect matchings M and M' with $|M \cap R| = p$ and $|M' \cap R| = q$. So we have $|R| = p + q - |(M \cap M') \cap R|$; $|R|$ will be minimum if we maximize the third term. We have $|(M \cap M') \cap R| \leq n - |\mathcal{C}|/2$ where \mathcal{C} is any family of vertex disjoint alternating cycles with respect to some perfect matching with $|\mathcal{C}|/2 \geq q - p$; taking such a family \mathcal{C} , with $|\mathcal{C}|$ minimum, will give the largest value of $n - |\mathcal{C}|/2$, so R has minimum cardinality. ■

Notice that if $p \geq q - 2$, we can use a single alternating cycle C instead of the family \mathcal{C} , since in any alternating cycle C , we have $|C|/2 \geq 2 \geq q - p$.

Corollary 3.3. *Let $G = (X, Y, E)$ with $|X| = |Y| = n$ be a Δ -regular bipartite graph, and let $\mathcal{P} = \{q - a, q\}$ with $1 \leq q \leq n$ and $1 \leq a \leq 2$. The minimum cardinality of a \mathcal{P} -feasible set R is given by*

$$|R| = q + \max\{0, q - n + |C|/2 - a\}$$

where C is a shortest cycle which is alternating with respect to some perfect matching in G .

Surprisingly the complexity of finding in a graph G a shortest possible alternating cycle with respect to some maximum matching (not given) is unknown even if G is a 3-regular bipartite graph. For reference purposes, this problem will be called the SAC problem (Shortest Alternating Cycle); it is formally defined as follows.

Instance: A graph $G = (V, E)$ and a positive integer $L \leq |V|$.

Question: Is there a maximum matching M and a cycle C with $|C| \leq L$ and $|C \cap M| = \frac{1}{2}|C|$?

Notice that the problem is easy if either a cycle C or a perfect matching M is given.

In order to give a sufficient condition for a regular graph (X, Y, E) with $|X| = |Y| = n$ to have a \mathcal{P} -feasible set R with $|R| = n + 1$ for $\mathcal{P} = \{0, 1, \dots, n\}$, we will need some preliminaries.

Lemma 3.4. *For any collection of n subsets S_1, \dots, S_n of a set $S = \{s_1, \dots, s_n\}$ such that $|S_i| = r \geq \sqrt{n} + 1, 1 \leq i \leq n$, there exist two subsets S_i and S_j such that $|S_i \cap S_j| \geq 2$.*

Proof: Assume we have a collection of m subsets S_i of S with $|S_i| = r$ for all $i \leq m$, and $|S_i \cap S_j| \leq 1$ for all $i, j \leq m$, then an element $s \in S$ is contained in at most $\frac{n-1}{r-1}$ subsets S_i (since the number of subsets S_i which contain s and which are otherwise disjoint is $\leq (n-1)/(r-1)$); so the total number m of subsets is at most $\frac{n(n-1)}{r(r-1)}$ because we can take n different elements s and in doing this each set is counted r times.

Now if we have $m = n > \frac{n(n-1)}{r(r-1)}$, then there will be two subsets S_i and S_j with $|S_i \cap S_j| \geq 2$. The smallest r verifying $r(r-1) \geq n-1$ is $r = \frac{1}{2}(1 + \sqrt{4n-3}) < \sqrt{n} + 1$. ■

In the following, we denote by $\delta(G)$ the minimum degree of a graph G and by $N(x)$ the set of neighbors of x . Furthermore C_k will denote a cycle of length k .

Corollary 3.5. *Let $G = (X, Y, E)$ be a bipartite graph with $|X| = |Y| = n, n \geq 4$. If $\delta(G) \geq \sqrt{n} + 1$, then G contains a cycle of length four.*

Proof: If $\delta(G) \geq \sqrt{n} + 1$, it follows from Lemma 3.4 that there exist two vertices x and x' in X such that $|N(x) \cap N(x')| \geq 2$. Hence, for $y, y' \in N(x) \cap N(x')$, (x, y, x', y') is a cycle. ■

Fact 3.6. *Let n, p and q be three integers such that $0 \leq q \leq p \leq n$.*

Then $p + \sqrt{n-p} + 1 \geq q + \sqrt{n-q} + 1$.

Proof: If $p = q$, the result is obvious. Consider now the case $p > q$.

Then $p + \sqrt{n-p} + 1 \geq q + \sqrt{n-q} + 1$

$$\Leftrightarrow p + \sqrt{n-p} \geq q + \sqrt{n-q}$$

$$\Leftrightarrow p - q \geq \sqrt{n-q} - \sqrt{n-p}$$

$$\Leftrightarrow (p-q)(\sqrt{n-q} + \sqrt{n-p}) \geq p-q$$

$$\Leftrightarrow \sqrt{n-q} + \sqrt{n-p} \geq 1$$

This is necessarily true, as $0 \leq q < p \leq n$. ■

Fact 3.7. $\frac{n+2\lceil\frac{n}{4}\rceil+1}{2} \geq 2(\lceil\frac{n}{4}\rceil - 1) + \sqrt{n - 2(\lceil\frac{n}{4}\rceil - 1)} + 1$

Proof: $\frac{n+2\lceil\frac{n}{4}\rceil+1}{2} \geq 2(\lceil\frac{n}{4}\rceil - 1) + \sqrt{n - 2(\lceil\frac{n}{4}\rceil - 1)} + 1$

$$\Leftrightarrow n + 2\lceil\frac{n}{4}\rceil + 1 \geq 4\lceil\frac{n}{4}\rceil - 4 + 2\sqrt{n - 2(\lceil\frac{n}{4}\rceil - 1)} + 2$$

$$\Leftrightarrow n - 2\lceil\frac{n}{4}\rceil + 3 \geq 2\sqrt{n - 2(\lceil\frac{n}{4}\rceil - 1)}$$

$$\Leftrightarrow n^2 + 4\lceil\frac{n}{4}\rceil^2 + 9 - 4n\lceil\frac{n}{4}\rceil - 12\lceil\frac{n}{4}\rceil + 6n \geq 4n - 8\lceil\frac{n}{4}\rceil + 8$$

$$\Leftrightarrow n^2 + 4\lceil\frac{n}{4}\rceil^2 + 1 - 4n\lceil\frac{n}{4}\rceil - 4\lceil\frac{n}{4}\rceil + 2n \geq 0$$

$$\Leftrightarrow 4\lceil\frac{n}{4}\rceil \leq 2n + 1 + (n - 2\lceil\frac{n}{4}\rceil)^2$$

Notice that $4\lceil\frac{n}{4}\rceil < n + 4 \leq 2n$ if $n \geq 4$, thus $4\lceil\frac{n}{4}\rceil \leq 2n + 1 + (n - 2\lceil\frac{n}{4}\rceil)^2$. \blacksquare

Lemma 3.8. *Let $G = (X, Y, E)$ be a simple bipartite graph with $|X| = |Y| = n$, $n \geq 4$. A sufficient condition for G to contain k vertex disjoint cycles of length four is*

$$\delta(G) - 2(k - 1) \geq \sqrt{n - 2(k - 1)} + 1$$

Proof: Let G_0, G_1, \dots, G_{k-1} be a sequence of graphs built as follows: $G_0 = G$ and G_i is the subgraph of G_{i-1} obtained by deleting four vertices $x, x' \in X$ and $y, y' \in Y$.

We have $\delta(G_i) \geq \delta(G_{i-1}) - 2$ and $n_i = n_{i-1} - 2$ where n_i is the number of vertices of the left set (or of the right set) in G_i .

From fact 3.6, if $1 \leq i \leq k$, we have:

$$\delta(G) \geq 2(k - 1) + \sqrt{n - 2(k - 1)} + 1 \geq 2(i - 1) + \sqrt{n - 2(i - 1)} + 1. \text{ As a consequence we have } \delta(G_{i-1}) \geq \delta(G) - 2(i - 1) \geq \sqrt{n - 2(i - 1)} + 1 = \sqrt{n_{i-1}} + 1 \text{ for all } i (1 \leq i \leq k).$$

So from Corollary 3.5, G_{i-1} contains a C_4 for all $i (1 \leq i \leq k)$ and by the construction we have found k vertex disjoint cycles C_4 (taking at each step four vertices x, x', y, y' forming a C_4). \blacksquare

Theorem 3.9. *Let $G = (X, Y, E)$ be a Δ -regular simple bipartite graph with $|X| = |Y| = n \geq 4$ and $\Delta \geq \frac{1}{2}(n + 2\lceil\frac{n}{4}\rceil + 1)$. Let $\mathcal{P} = \{0, 1, \dots, n\}$; then there exists a \mathcal{P} -feasible set R with $|R| = n + 1$.*

Proof: We have $\Delta \geq \frac{1}{2}(n + 2\lceil\frac{n}{4}\rceil + 1) \geq 2(\lceil\frac{n}{4}\rceil - 1) + \sqrt{n - 2(\lceil\frac{n}{4}\rceil - 1)} + 1$ from Fact 3.7. It follows from Lemma 3.8 that G contains $\lceil n/4 \rceil$ vertex disjoint cycles C_4 . Let $\{x_{2i+1}, y_{2i+1}, x_{2i+2}, y_{2i+2}\}$ be the vertices of cycle C_4^i for $i = 0, \dots, \lceil n/4 \rceil - 1$.

We observe that the number of vertices of X (or of Y) contained in the cycles C_4^i is $2\lceil n/4 \rceil \geq n - \lceil n/2 \rceil = \lfloor n/2 \rfloor$.

Let now $H = (X', Y', E')$ be the subgraph of G obtained by deletion of all the cycles C_4^i and their vertices. We have $|X'| = |Y'| = n - 2\lceil n/4 \rceil \leq \lfloor n/2 \rfloor$ and $\delta(H) \geq \Delta - 2\lceil n/4 \rceil \geq \frac{1}{2}(n + 2\lceil n/4 \rceil + 1) - 2\lceil n/4 \rceil = \frac{1}{2}(n - 2\lceil n/4 \rceil + 1) = \frac{1}{2}(|X'| + 1)$.

It is known (see [4] [Corollary 7.3.13]) that such an H is Hamiltonian. It has then a 2-factor which can be partitioned into two perfect matchings M_H, M'_H of H . Notice that $|M_H| = |M'_H| = n - 2\lceil n/4 \rceil$.

We now construct R as follows:

$$R = \{[x_i, y_i] \mid i = 1, \dots, 2\lceil n/4 \rceil\} \cup \{[x_1, y_2]\} \cup M_H.$$

Clearly we construct perfect matchings in G by taking a perfect matching in each C_4^i and in H . Each C_4^i ($i \geq 1$) will give matchings with 0 or 2 edges of R ; C_4^0 will give matchings with 1 or 2 edges of R . In H , the matchings M_H and M'_H have $|M_H|$ or 0 edges in R .

From the cycles C_4^i we can construct matchings having $1, 2, \dots, 2\lceil n/4 \rceil$ edges in R . These can be combined with $M_{H'}$ to get perfect matchings M_i in G having $1, 2, \dots, 2\lceil n/4 \rceil$ edges in R . Combining these matchings with M_H will give perfect matchings in G having $1 + |M_H|, 2 + |M_H|, \dots, 2\lceil n/4 \rceil + |M_H|$ edges in R . Since $|M_H| = n - 2\lceil n/4 \rceil \leq \lceil n/2 \rceil$ we will produce perfect matchings of G having i edges in R for any i with $1 \leq i \leq n$.

Now since G is regular, we may remove the edges of all cycles C_4^i and of $M_H \cup M'_H$. We have a $(\Delta - 2)$ -regular graph, which has a perfect matching M_0 such that $M_0 \cap R = \emptyset$. So we have constructed a \mathcal{P} -feasible set R with $|R| = 2\lceil n/4 \rceil + 1 + |M_H| = n + 1$. ■

The following is a simple consequence of the König theorem.

Proposition 3.10. *If a 3-regular bipartite graph contains a cycle on four vertices, then this cycle is alternating with respect to some perfect matching.*

Remark 3.11. *In general a graph G may not have perfect matchings. We can find a minimum cardinality \mathcal{P} -feasible set R for $\mathcal{P} = \{n - 1, n\}$. Here M_n is a maximum matching which is not perfect; an alternating chain $C = \{e_1, e_2\}$ exists which starts at some (exposed) vertex. We simply remove e_2 from M_n and introduce e_1 into M_n to obtain M_{n-1} . So $R = M_n \cup \{e_1\} - \{e_2\}$.*

Let us mention additional results related to alternating cycles in bipartite graphs.

Theorem 3.12. *Let $G = (X, Y, E)$ be a Δ -regular bipartite graph ($\Delta \geq 3$) with $|X| = |Y| = n$, then G contains a cycle C with $|C| \leq 2\lceil n/2 \rceil$ which is alternating with respect to some perfect matching.*

Proof: Let $(M_1, M_2, \dots, M_\Delta)$ be an edge Δ -coloring of G ; if $M_1 \cup M_2$ is not a Hamiltonian cycle, then it contains a cycle C with $|C| \leq 2\lceil n/2 \rceil$. C is clearly alternating for M_1 .

If $M_1 \cup M_2$ is a Hamiltonian cycle, then consider any edge $e \in M_3$; $M_1 \cup M_2 \cup \{e\}$ contains 2 cycles using e ; at least one of them has at most $2\lceil n/2 \rceil$ edges; this cycle is alternating with respect to M_1 or M_2 . ■

Theorem 3.13. *Let G be a Δ -regular bipartite graph (with $\Delta \geq 3$) such that for some integer $k \geq 3$ every cycle of length at least $2k$ has a chord. Then there exists a cycle C with $|C| \leq 2k - 2$ which is alternating with respect to some perfect matching.*

Proof: Take a perfect matching M in G ; since $\Delta \geq 3$ from the König theorem, there exists a perfect matching M' with $M' \cap M = \emptyset$. Then $M' \cup M$ contains a cycle C which is alternating with respect to M . Assume $|C| \geq 2k$; then there is a chord $[a, b]$. It determines with one part of C an alternating cycle C' with respect to M . Now $|C'| \leq |C| - 2$. If $|C'| \geq 2k$ we continue. We will finally get a cycle C'' with $|C''| \leq 2k - 2$ which will be alternating with respect to M . ■

A tedious but not difficult enumeration of cases shows the following.

Theorem 3.14. *For a 3-regular bipartite graph $G = (X, Y, E)$ with $|X| = |Y| = n \leq 7$, there exists a set $R \subseteq E$ with $|R| \leq n + 2$ which is \mathcal{P} -feasible for $\mathcal{P} = \{0, 1, \dots, n\}$.*

This result is best possible in the sense that there exists a bipartite 3-regular graph on $2n = 14$ vertices for which the minimum value of $|R|$ is $n + 2 = 9$; this is the so-called Heawood graph (or (3,6)-cage) (see [38], p.309).

In 3-regular bipartite graphs $G = (X, Y, E)$ with $|X| = |Y| = n \geq 8$ the minimum cardinality of a \mathcal{P} -feasible set R for $\mathcal{P} = \{0, 1, \dots, n\}$ is not known.

Finally if we restrict \mathcal{P} to $\{0, 1, \dots, p\}$ with $p \leq 4$, we can state the following.

Theorem 3.15. *Let $p \leq 4$ be an integer. For a 3-regular bipartite graph $G = (X, Y, E)$, with $|X| = |Y| = n \geq 2(p - 1)$, there exists a set $R \subseteq E$ with $|R| = p$ which is \mathcal{P} -feasible for $\mathcal{P} = \{0, 1, \dots, p\}$.*

Proof: We just give the proof for $p = 4$; the case $p \leq 3$ can be handled similarly. Let M and M' be two disjoint perfect matchings in G . So $E - (M \cup M')$ is also a perfect matching. Suppose $M \cup M'$, which is a 2-factor, is connected. It is then a Hamiltonian cycle \mathfrak{C} of G . Choose two chords of \mathfrak{C} , say $[a, b]$ and $[c, d]$, which are at distance at least 2 (i.e. there are no two vertices of the chords that are adjacent). These chords, belonging to $E - (M \cup M')$, divide the set of edges of cycle \mathfrak{C} into four parts A, B, C and D .

Let M_1 (resp. M_2) be the matching containing the chord $[a, b]$ (resp. $[c, d]$) and $|M| - 1$ edges of \mathfrak{C} . M_1 and M_2 use the same edges of $M \cup M'$ in two parts, say A and C , and different edges of $M \cup M'$ in B and D . As the chords are at distance at least 2, there are at least two edges $e_1, e_2 \in M_1 \cap M_2$. Let us distinguish two cases:

- (1) There exist $e_1, e_2 \in M_1 \cap M_2 \cap M$ (if necessary exchange M and M').

It is obvious that there exists an edge e_3 in $B \cup D$ such that $e_3 \in M_1 \cap M$ or $e_3 \in M_2 \cap M$ as M_1 and M_2 use different edges in B and D . Suppose there exists $e_3 \in M_1 \cap M$. Take $R = \{[a, b], e_1, e_2, e_3\}$. Then we have: $|M \cap R| = 3$, $|M' \cap R| = 0$, $|M_1 \cap R| = 4$, $|M_2 \cap R| = 2$ and $|(E - (M \cup M')) \cap R| = 1$.

- (2) There exist no two edges e_1 and e_2 such that $e_1, e_2 \in M_1 \cap M_2 \cap M$ or $e_1, e_2 \in M_1 \cap M_2 \cap M'$.

This of course implies $|A| = |C| = 3$. Suppose $e_1 \in M_1 \cap M_2 \cap M$ and $e_2 \in M_1 \cap M_2 \cap M'$. It is obvious that there exist two edges e_3 and e_4 in $B \cup D$ such that $e_3, e_4 \in M_1 \cap M$ or $e_3, e_4 \in M_2 \cap M$ as at least one part B or D contains at least four edges. Suppose $e_3, e_4 \in M_1 \cap M$. Take $R = \{e_1, e_2, e_3, e_4\}$. Then we have $|M \cap R| = 3$, $|M' \cap R| = 1$, $|M_1 \cap R| = 4$, $|M_2 \cap R| = 2$ and $|(E - (M \cup M')) \cap R| = 0$.

Suppose now $M \cup M'$ is not connected. Then it consists of at least two cycles. Notice that if there are more than two cycles, the solution is obvious. In fact consider three cycles C^1 , C^2 and C^3 . Take $R = \{e_1, e_2, e_3, e_4\}$ such that $e_1, e_2 \in C^1 \cap M$, $e_3 \in C^2 \cap M$ and $e_4 \in C^3 \cap M'$. Then of course there exist perfect matchings with 0, 1, 2, 3 and 4 edges of R . So suppose now that $M \cup M'$ consists of exactly two cycles, C^1 and C^2 . Distinguish two cases:

- (1) At least one of the cycles, say C^1 , has a chord $[a, b]$.

This chord divides cycle C^1 in two parts A and B which contain both an odd number of edges. Let M_1 be the perfect matching containing chord $[a, b]$, the edges of M in A , the edges of M' in B (if necessary exchange M and M') and edges of M in cycle C^2 . Let M_2 be the perfect matching containing chord $[a, b]$, the edges of M_1 in C^1 and edges of M' in cycle C^2 . So there exist at least two edges $e_1, e_2 \in M_1 \cap M_2$ such that $e_1 \in M$ and $e_2 \in M'$. Consider $e_3, e_4 \in C^2$ such that $e_3, e_4 \in M$. Take $R = \{e_1, e_2, e_3, e_4\}$. Then we have $|M \cap R| = 3$, $|M' \cap R| = 1$, $|M_1 \cap R| = 4$, $|M_2 \cap R| = 2$ and $|(E - (M \cup M')) \cap R| = 0$.

- (2) None of the two cycles has a chord.

This is only possible if both cycles have same length. As $n \geq 6$, we have $|C^1| = |C^2| \geq 6$. Consider two edges $[a, b], [c, d] \in E - (M \cup M')$ such that vertices a and c are neighbors in C^1 . Let M_1 be the perfect matching containing $[a, b], [c, d]$, edges of M in C^1 and edges of $M \cup M'$ in C^2 . Let $e_1, e_2 \in C^1 \cap M_1$ and $e_3 \in C^2 \cap M_1 \cap M$. Take $R = \{[a, b], e_1, e_2, e_3\}$. Then we have $|M_1 \cap R| = 4$, $|M' \cap R| = 0$, $|M \cap R| = 3$, taking M in C^1 and M' in C^2 , we obtain a perfect matching with two edges of R , taking M' in C^1 and M in C^2 , we obtain a perfect matching with one edge in R .

■

3.2 The interval property (IP)

We shall consider here the special case where \mathcal{P} is a set of consecutive integers and we will characterize graphs which have a property related to such a \mathcal{P} . We will exhibit some classes of graphs (bipartite or not) for which a \mathcal{P} -feasible set R with minimum cardinality can be obtained in polynomial time. We will denote by $\nu(G)$ the cardinality of a maximum matching in G .

We shall say that G has property IP (interval property) if whenever there are maximum

matchings M_k and M_ν in G with $|M_k \cap M_\nu| = k < \nu = \nu(G)$, there are also maximum matchings M_i with $|M_i \cap M_\nu| = i$ for $i = k, k+1, \dots, \nu$.

In other words, when G has property IP and there is some k and two maximum matchings M_k and M_ν with $|M_k \cap M_\nu| = k \leq \nu(G)$, then $R = M_\nu$ is \mathcal{P} -feasible for $\mathcal{P} = \{k, k+1, \dots, \nu = \nu(G)\}$ and R has minimum cardinality.

We define a *IP-perfect* graph G as a graph in which every partial subgraph has property IP. We remind the reader that a partial subgraph of a graph G is obtained by taking an induced subgraph G' and keeping only a subset of edges of G' .

A *cactus* is a graph where any two (elementary) cycles have at most one common vertex. A cactus is *odd* if all its (elementary) cycles are odd. Notice that a tree is a special (odd) cactus.

Theorem 3.16. *The following statements are equivalent:*

- (a) G is an odd cactus;
- (b) G is IP-perfect.

Proof: (b) \Rightarrow (a). If G is not an odd cactus, there exists a partial subgraph which is an even cycle $C = M_\nu \cup M_0$ with $|M_\nu| = |M_0| \geq 2$. We have $M_0 \cap M_\nu = \emptyset$ by construction and $|M_\nu \cap M_\nu| \geq 2$, but there is no M_1 with $|M_1| = |M_\nu|$ and $|M_1 \cap M_\nu| = 1$.

(a) \Rightarrow (b). Assume we have two maximum matchings M_k and M_ν with $|M_k \cap M_\nu| = k < \nu = \nu(G)$. Consider $M_k \Delta M_\nu$. It consists of a collection of vertex disjoint even alternating chains with total length $2(\nu - k) > 0$.

We may use some of the subchains (by starting from the endvertices saturated by M_ν) to replace r edges of M_ν by r edges of M_k in order to obtain a maximum matching M_{k+r} with $|M_{k+r} \cap M_\nu| = k+r$ for $r = 1, \dots, \nu(G) - k$. Since every partial subgraph of G is also an odd cactus, G has property IP. ■

It follows from Theorem 3.16 that if we want to find the largest sequence of consecutive integers $\mathcal{P} = \{p_0, p_1, \dots, p_s\}$ such that a set $R = M_\nu$ is \mathcal{P} -feasible for an odd cactus G , we have to find in G two maximum matchings M_k and M_ν such that $|M_k \cap M_\nu|$ is minimum.

Let us examine first the case of bipartite graphs (that include trees but not odd cacti).

Theorem 3.17. *If $G = (X, Y, E)$ is a bipartite graph, there exists a polynomial time algorithm to construct two maximum matchings M, M' with a minimum value of $|M \cap M'|$.*

Proof: Let us replace each edge $[x, y]$ of G by two arcs $(x, y)^0$ and $(x, y)^1$ with capacities $c(x, y)^0 = c(x, y)^1 = 1$ and costs $k(x, y)^0 = 0$, $k(x, y)^1 = 1$. Introduce a source s with arcs (s, x) having $c(s, x) = 2$ and $k(s, x) = 0$ for each vertex x in X . Similarly for each vertex y in Y introduce a sink t with arcs (y, t) with $c(y, t) = 2$ and $k(y, t) = 0$.

Construct in the network N obtained in this way an (integral) maximum flow f from s to t with minimum cost $K(f)$. There exists a feasible flow with value $2\nu(G)$ (obtained by setting

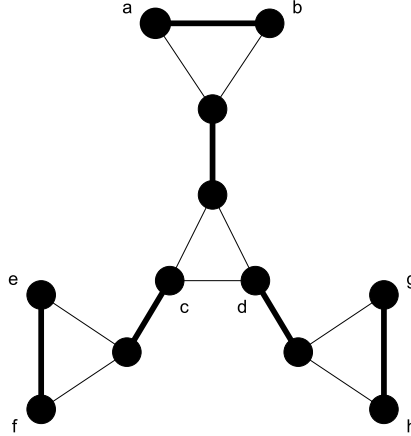


Figure 3.1: An odd cactus where the 2-matching algorithm does not give the solution.

$f(x, y)^1 = f(x, y)^0 = 1$ for all arcs (x, y) corresponding to the edges $[x, y]$ of a maximum matching in G). Furthermore no flow can have a value larger than $2\nu(G)$ (because this would mean that there is in G a matching M with $|M| > \nu(G)$). Now the cost of f is equal to the number of arcs $(x, y)^1$ with $f(x, y)^1 = 1$. Since $K(f)$ has been minimized, we have a minimum number of such arcs and furthermore $f(x, y)^1 = 1$ implies $f(x, y)^0 = 1$. These are the edges $[x, y]$ of G which are used in both matchings M and M' . Hence an integral flow f with maximum value $2\nu(G)$ and minimum cost $K(f)$ will define two maximum matchings M and M' with $|M \cap M'| = K(f)$ minimum.

It is known that such a flow can be constructed in polynomial time (see [1]). ■

Remark 3.18. For non bipartite graphs, one cannot use the same construction (duplication of edges) and determination of a maximum 2-matching (partial graph H with degrees $d_H(z) \leq 2$ for each vertex z).

In the graph of Figure 3.1, we would obtain a 2-matching H consisting of the edges of all four triangles; its cost is 0. It is clearly not the union of two maximum matchings. The two maximum matchings M and M' with $|M \cap M'|$ minimum are $M = M'$ given by the heavy edges; the cost of this 2-matching is 6.

At this stage, we can deduce from Theorems 3.16 and 3.17.

Theorem 3.19. If $G = (V, E)$ is a forest, we can determine in polynomial time a minimum k and a minimum set R of edges to be colored in red in such a way that for $i = k, k + 1, \dots, \nu(G)$, G has a maximum matching M_i with $|M_i \cap R| = i$.

Remark 3.20. In a graph G with the IP property, there exists a set R with $|R| = \nu(G)$ such that for $i = 0, 1, \dots, \nu(G)$ G has a maximum matching M_i with $|M_i \cap R| = i$ if and only if G has two disjoint maximum matchings.

It should be noticed that finding in a graph two maximum matchings that are as disjoint as possible is \mathcal{NP} -complete. This is an immediate consequence of the \mathcal{NP} -completeness of deciding whether a 3-regular graph has an edge 3-coloring [43].

We will now show that there is an algorithm to determine if some special odd cacti have two disjoint maximum matchings.

In [31], it is shown that finding a maximum number of edges that can be colored with 2 colors is \mathcal{NP} -hard in multigraphs.

Hartvigsen has developed (see [42]) an algorithm for constructing in a graph a partial graph H with $d_H(v) \leq 2$ for each vertex v , which contains no triangle and which has a maximum number of edges.

Such an algorithm can be used in graphs where the only odd cycles are triangles (these are the so called line-perfect graphs (see [24], [71])). We obtain the following.

Theorem 3.21. *If G is a line-perfect graph, one can determine in polynomial time whether G contains two disjoint maximum matchings.*

Proof: We apply the algorithm of Hartvigsen that gives a partial graph H with $d_H(v) \leq 2$ for each v , which contains no triangle and which has a maximum number $|E(H)|$ of edges. Since G has no odd cycle of length 5 or more, $E(H)$ has no odd cycle and is therefore the union of two disjoint matchings M_1 and M_2 .

We cannot have $|E(H)| > 2\nu(G)$ because this would imply that H contains a matching M with $|M| > \nu(G)$. So we have $|E(H)| \leq 2\nu(G)$ and if $|E(H)| < 2\nu(G)$, then clearly G cannot contain two disjoint maximum matchings. So assume we have $|E(H)| = 2\nu(G)$. Since $|M_1|, |M_2| \leq \nu(G)$ and $|E(H)| = 2\nu(G) = |M_1| + |M_2|$, we have two disjoint matchings M_1 and M_2 with $|M_1| = |M_2| = \nu(G)$. ■

From Theorems 3.16 and 3.21 we obtain the following.

Corollary 3.22. *If G is a cactus where all cycles are triangles, one can determine in polynomial time whether there exists a minimum set R of edges that is \mathcal{P} -feasible for $\mathcal{P} = \{0, 1, \dots, \nu(G)\}$.*

It should be noted that in order to find two maximum matchings M and M' with $|M \cap M'|$ minimum, we would need to introduce weights on the edges, but apparently this cannot be handled by the above algorithm.

3.3 Conclusion

We have examined the problem of finding a minimum subset R of edges for which there exist maximum matchings M_i with $|M_i \cap R| = p_i$ for some given values of p_i . Partial results have been obtained for some classes of graphs (regular bipartite graphs, forests, odd cacti

with triangles only, ...). In general, our problem requires the determination of a shortest alternating cycle (SAC problem) whose complexity status is open. Further research is needed to extend our results to other classes of graphs. In particular the case of general odd cacti would be interesting to analyze since these are exactly the IP-perfect graphs.

These problems seem to be more difficult than the spanning tree problems in bicolored graphs mentioned in the introduction; the reason is that it is a special case of three matroid intersections as mentioned in [34]: a matching is an intersection problem of two matroids and the bicoloring (R, B) induces a partition matroid; for trees we simply have, in addition to the partition matroid, a second matroid whose independent sets are the forests in G . Such problems are known to be solvable in polynomial time (see [56]).

Chapter 4

On a graph coloring problem arising from discrete tomography

Introduction

In Chapter 1, we introduced the field of discrete tomography which deals with the reconstruction of discrete objects from their projections.

Here we shall consider a graph coloring problem which generalizes the basic image reconstruction problem in discrete tomography (see Chapter 1).

We are given a connected graph $G = (V, E)$ and a collection \mathcal{P} of p subsets P_i of vertices of G . We are also given a set $\{1, 2, \dots, k\}$ of colors as well as a collection H of p vectors $h(P_i) = (h_i^1, \dots, h_i^k) \in \mathbb{N}^k$ ($i = 1, \dots, p$).

We have to find a k -partition V^1, V^2, \dots, V^k of V such that

$$|P_i \cap V^j| = h_i^j \text{ for all } i \leq p \text{ and all } j \leq k. \quad (4.1)$$

This problem will be called $A(G, k, \mathcal{P}, H)$. It is clear that in this formulation the structure of G plays no role.

We shall from now on consider a family of chains μ_i in G ; we will denote by P_i the (ordered) set of vertices in μ_i and the length of μ_i will be $|P_i|$. Whenever no confusion may arise, we shall identify μ_i with its vertex set P_i . We will then call $|P_i|$ the length of P_i . In the case where the structure of G plays no role, it is not restrictive to start from chains μ_i (instead of arbitrary subsets P_i of vertices as above): we can indeed link the vertices of a P_i to form a chain μ_i .

The k -partition need not be a coloring of G where adjacent vertices have different colors. We will talk indifferently of k -partition or k -coloring to describe a partition of the vertex set into k subsets (color classes); whenever we will have the usual requirement of having different colors on adjacent vertices, we will call this a *proper* k -coloring. The corresponding

reconstruction problem associated with proper k -colorings will be denoted $\Lambda^*(G, k, \mathcal{P}, H)$.

Let us recall the relation between this problem and the basic image reconstruction problem in discrete tomography. Consider the special case where $G = (V, E)$ is a grid graph; its vertex set is $V = \{x_{rs} \mid r = 1, \dots, m; s = 1, \dots, n\}$ and its edge set is

$$E = \{[x_{rs}, x_{r,s+1}] \mid s = 1, \dots, n-1; r = 1, \dots, m\} \cup \\ \{[x_{rs}, x_{r+1,s}] \mid r = 1, \dots, m-1; s = 1, \dots, n\}$$

If x_{rs} is located in row r and column s of the grid, then by taking for \mathcal{P} the collection of chains $P_r = \{x_{r1}, \dots, x_{rn}\}$ for $r = 1, \dots, m$, and $P_{m+s} = \{x_{1s}, \dots, x_{ms}\}$ for $s = 1, \dots, n$, $\Lambda(G, k, \mathcal{P}, H)$ is exactly the basic image reconstruction problem in discrete tomography; here h_r^j (resp. h_{m+s}^j) is the number of occurrences of color j in row r (resp. in column s) (i.e. (h_r^1, \dots, h_r^k) and $(h_{m+s}^1, \dots, h_{m+s}^k)$ are the horizontal and the vertical projections, respectively).

As already mentioned previously, the problem for $k = 2$ consists of reconstructing a $(0, 1)$ -matrix from its vertical and horizontal projections, i.e., number of occurrences of 1 in each row and in each column; this case is solved in polynomial time [65].

We recall that for $k = 4$, this problem is \mathcal{NP} -complete [16]; for $k = 3$ the complexity status is open but some special cases were solved in polynomial time (see Chapter 6 and [18, 19]).

In this chapter we will consider some extensions and variations of this basic problem by taking more general classes of graphs G such as trees, bipartite graphs, planar graphs, cacti.

As an application of $\Lambda(G, k, \mathcal{P}, H)$ let us mention the following problem consisting in scheduling the refurbishment of the stations in a city subway network. The network is represented by a graph $G = (V, E)$ where the vertices are the stations. Each metro line is associated with a chain P_i . Assuming that the renovation operation of every single station takes one month, we want to schedule these operations while taking into account the following requirements: in month j , the number of stations in metro line P_i which will be closed for renovation is h_i^j . The problem of assigning a date (month) for the renovation of every station with the above constraints is precisely $\Lambda(G, k, \mathcal{P}, H)$ if the whole refurbishment has to take place in a period of k months. In some cases, it is desired to avoid closing two consecutive stations along the same metro line; the assignment of dates is then a proper k -coloring of the underlying graph G and the problem is $\Lambda^*(G, k, \mathcal{P}, H)$.

In addition to the aforementioned application, our problem may be viewed in a different context related to constraint satisfaction in logic. Essentially we are given a collection of n Boolean variables as well as a collection of clauses P_i (each one of them involves a subset of the Boolean variables). It is required to find an assignment of values ‘true’ or ‘false’ to each Boolean variable in such a way that in each clause P_i the number of variables with value ‘false’ is exactly (or at most) a given number h_i^F . Notice that here we have a number k of colors which is 2. The general k -coloring case would then correspond to k -valued logical variables.

After preliminaries given in Section 4.1, we will consider the basic problem $\Lambda(G, k, \mathcal{P}, H)$ in Section 4.2 with the case $k = 2$ (difficult and easy cases) and the general case $k \geq 3$. Then Section 4.3 will be dedicated to the case of proper colorings, i.e., to $\Lambda^*(G, k, \mathcal{P}, H)$. In Section 4.4 we will consider line graphs. This amounts to replacing the vertex colorings by edge colorings. Again we will consider general k -colorings and also proper k -colorings. Finally, Section 4.5 will present a summary of the results obtained in this chapter.

4.1 Preliminaries

In the following we assume that several basic conditions for a solution to exist are verified, in particular $\sum_{j=1}^k h_i^j = |P_i|$, for all $i = 1, \dots, p$. In addition, if we want to determine proper colorings, we have to assume that $h_i^j \leq \lceil \frac{|P_i|}{2} \rceil$ for all i, j . It follows that there is at most one color such that $h_i^j = \lceil \frac{|P_i|}{2} \rceil$ if $|P_i|$ is odd and at most two colors such that $h_i^j = \frac{|P_i|}{2}$ if $|P_i|$ is even. These colors will be called *saturating* for P_i .

We need some more definitions and notations for \mathcal{P} . For a family $\mathcal{P} = (P_i \mid i = 1, \dots, p)$ of subsets P_i of a set V , we call *cover index* of \mathcal{P} and denote by $c(\mathcal{P})$ the maximum number of members of \mathcal{P} which may cover a single element of V (i.e., which have a non empty intersection).

For instance in the basic image reconstruction problem of discrete tomography we have $c(\mathcal{P}) = 2$.

A family $\mathcal{P} = (P_i \mid i = 1, \dots, p)$ of subsets P_i of a set V is called *nested* if for any $P_i, P_f \in \mathcal{P}$, we have either $P_i \subseteq P_f$ or $P_f \subseteq P_i$ or $P_i \cap P_f = \emptyset$.

Consider now a partition of \mathcal{P} into nested families. One can look for a partition into the smallest possible number of nested families. This number, denoted by $\text{Nest}(\mathcal{P})$, is called the *nesticity* of \mathcal{P} .

Fact 4.1. [39] *One can determine in polynomial time if for a family \mathcal{P} we have $\text{Nest}(\mathcal{P}) \leq 2$.*

Proof: Assign a vertex to each $P_i \in \mathcal{P}$ and link by an edge P_i and P_f whenever $P_i \cap P_f \neq \emptyset$, $P_i \not\subseteq P_f$ and $P_f \not\subseteq P_i$. The resulting graph is bipartite if and only if $\text{Nest}(\mathcal{P}) \leq 2$. ■

Observe that $c(\mathcal{P})$ and $\text{Nest}(\mathcal{P})$ are unrelated: we may have $c(\mathcal{P}) > \text{Nest}(\mathcal{P})$ or $c(\mathcal{P}) < \text{Nest}(\mathcal{P})$. For example, for $\mathcal{P} = (\{a, b\}, \{a, c\}, \{b, c\})$, we have $c(\mathcal{P}) = 2$ and $\text{Nest}(\mathcal{P}) = 3$, and for $\mathcal{P}' = (\{a, b, c\}, \{a, b\})$, we have $c(\mathcal{P}') = 2$ and $\text{Nest}(\mathcal{P}') = 1$.

4.2 Arbitrary colorings

In this section we establish some complexity results and we exhibit some cases which can be solved in polynomial time for $\Lambda(G, k, \mathcal{P}, H)$.

Notice that whenever the k -colorings are not required to be proper, we can assume that for each edge e there is at least one chain μ_i which uses e ; otherwise the edge can be removed. Notice that it may happen that we get a disconnected graph; in such a case the problem is decomposed.

We shall start with the case where we have $k = 2$ colors.

4.2.1 Difficult problems for $k = 2$

Let us first give two statements which do not refer to the nature of the underlying graph G .

Theorem 4.2. $\Lambda(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete if \mathcal{P} is a 3-uniform family ($|P_i| = 3$ for $i = 1, \dots, p$) which is 3-regular (each vertex is in exactly three P_i 's).

Proof: We use a transformation from the CUBIC PLANAR MONOTONE 1-in-3SAT problem which is known to be \mathcal{NP} -complete (see [60]). In this problem we are given a set X of variables and a set C of clauses of the form $(x \vee y \vee z)$ where x, y and z are distinct variables without negation such that the underlying bipartite graph $G = (X \cup C, E) = (X \cup C, \{[x_i, \hat{c}] | x_i \text{ occurring in clause } \hat{c} \in C\})$ is 3-regular and planar. The question is to decide whether there exists a truth assignment such that exactly one variable in each clause is 'true'.

We associate with each clause $c = (x \vee y \vee z)$ a chain $P_c = \{x, y, z\}$. Since the underlying bipartite graph G is 3-regular we have that each vertex x is in exactly three chains. We set $h(P_c) = (1, 2)$ for each chain P_c .

If an instance of CUBIC PLANAR MONOTONE 1-in-3SAT has answer 'yes', then by setting $V^1 = \{x | x \text{ is true}\}$ we get an positive answer for $\Lambda(G, 2, \mathcal{P}, H)$. Conversely if $\Lambda(G, 2, \mathcal{P}, H)$ has a positive answer then by setting each variable x such that $x \in V^1$ to 'true', we will get a positive answer for CUBIC PLANAR MONOTONE 1-in-3SAT. ■

Theorem 4.3. $\Lambda(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete if $\text{Nest}(\mathcal{P}) = 3$.

Proof: We use a transformation of the 3-dimensional matching problem which is known to be \mathcal{NP} -complete [36]. To state a 3-dimensional matching problem, we introduce a collection of points with coordinates (α, β, γ) with $\alpha, \beta, \gamma \in \{1, 2, \dots, q\}$ and 3 families formed by all disjoint chains parallel to the coordinate axes; this gives \mathcal{P} with $\text{Nest}(\mathcal{P}) = 3 = c(\mathcal{P})$.

We set $h_i^1 = 1, h_i^2 = |P_i| - 1$ for each P_i in \mathcal{P} . Then there exists a matching of size q if and only if there exists a partition V^1, V^2 of the set of points which satisfies (4.1). ■

Notice that it follows from this transformation that $\Lambda(G, k, \mathcal{P}, H)$ remains \mathcal{NP} -complete for $k = 2$ and $c(\mathcal{P}) = 3$.

Theorem 4.4. $\Lambda(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete when G is bipartite of maximum degree ≤ 4 and each color occurs at most 3 times in each P_i ($h_i^j \leq 3, \forall i = 1, \dots, p, \forall j = 1, 2$) and $|P_i \cap P_f| \leq 1$ for all $1 \leq i, f \leq p$ ($i \neq f$).

Proof: The transformation is from the \mathcal{NP} -complete problem ONE-IN-THREE 3SAT which is defined as follows [36].

Instance: A set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$ variables.

Question: Is there a truth assignment for U such that each clause in C has exactly one true literal?

This problem is also \mathcal{NP} -complete in the case where there is no negated literal.

We build a graph by associating with each variable x occurring s times vertices $x_1, x_{12}, x_2, x_{23}, x_3, \dots, x_{s-1,s}, x_s$ and edges $[x_1, x_{12}], [x_{12}, x_2], [x_2, x_{23}], \dots, [x_{s-1,s}, x_s]$. For each clause $c_l = \{x, y, z\}$ we know the number of occurrences of its variables in clauses c_1, \dots, c_{l-1} ; so assume $c_l = \{x_d, y_e, z_f\}$ which means that in c_l , x has its d^{th} occurrence, y its e^{th} occurrence and z its f^{th} . We introduce vertices u_l and w_l with edges $[x_d, u_l], [u_l, y_e], [y_e, w_l], [w_l, z_f]$. Clearly the graph obtained is bipartite. Now we define \mathcal{P} .

For each variable x , each edge $[x_1, x_{12}], [x_{12}, x_2], \dots, [x_{s-1,s}, x_s]$ becomes a chain P'_i with $h(P'_i) = (1, 1)$. For each clause $c_l = \{x_d, y_e, z_f\}$ we introduce a chain $P''_i = \{x_d, u_l, y_e, w_l, z_f\}$ with $h(P''_i) = (3, 2)$ and also chains $P_l^* = \{u_l\}, P_l^{**} = \{w_l\}$ with $h(P_l^*) = h(P_l^{**}) = (1, 0)$. The family \mathcal{P} of chains obtained verifies clearly $|P_i \cap P_f| \leq 1$ for all $i, f \leq p$ ($i \neq f$). Furthermore no vertex of G has degree more than 4.

If an instance of ONE-IN-THREE 3SAT has answer ‘yes’, then assigning color 1 to vertices u_l, w_l (for all l) and to x_1, x_2, \dots, x_s if variable x is ‘true’, or to $x_{12}, x_{23}, \dots, x_{s-1,s}$ otherwise, and assigning color 2 to the remaining vertices gives a positive answer to the corresponding instance $\Lambda(G, 2, \mathcal{P}, H)$. Conversely if an instance of $\Lambda(G, 2, \mathcal{P}, H)$ is positive, then all vertices u_l, w_l (for all l) have color 1, so for each chain $P''_i = \{x_d, u_l, y_e, w_l, z_f\}$ there is exactly one vertex in $\{x_d, y_e, z_f\}$ with color 1. Furthermore from the requirements on the chains P'_i , for each variable x , all vertices x_1, x_2, \dots, x_s have the same color. So assigning the value ‘true’ to x if x_1, x_2, \dots, x_s have color 1, or value ‘false’ otherwise, we get a positive answer to ONE-IN-THREE 3SAT. \blacksquare

Theorem 4.5. $\Lambda(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete when G is a tree with maximum degree 3.

Proof: Again, we reduce from the \mathcal{NP} -complete problem ONE-IN-THREE 3SAT with no negated literal, already defined. We denote by x_1, \dots, x_ν the variables, and by c_1, \dots, c_α the clauses. We construct a tree as follows. There is a main path Π with $\nu + \alpha$ vertices. Each one of the ν first vertices of Π is linked by an edge to a leaf, the i^{th} leaf being labelled by x_i (we shall speak of a *variable leaf*). Each one of the α next vertices of Π is linked to a *clause gadget* (so that, in our tree, there is one gadget for each clause): the gadget for a clause $c_h = x_i \vee x_j \vee x_k$ is a tree with five vertices (labelled a_h, b_h, x_i, x_j and x_k), x_i, x_j and x_k being the 3 leaves, and a_h being linked to Π by an edge. The edges inside the gadget are $[a_h, x_i], [a_h, b_h], [b_h, x_j]$ and $[b_h, x_k]$ (see Figure 4.1 for an example). Note that the tree constructed so far has maximum degree 3.

It remains to describe the collection \mathcal{P} . First, in the gadget of clause $c_h = x_i \vee x_j \vee x_k$, there

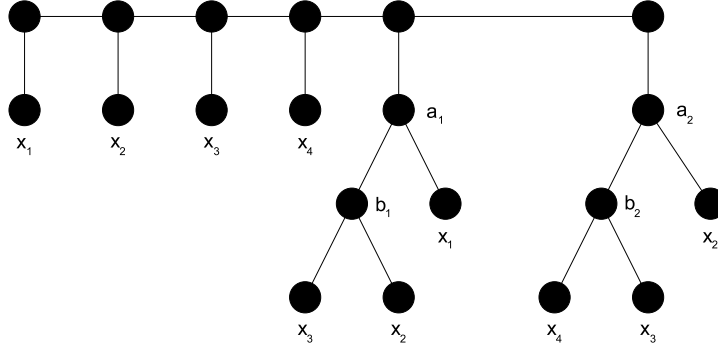


Figure 4.1: The tree constructed for the instance $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4)$.

is a chain $P_h = \{a_h\}$ with $h(P_h) = (1, 0)$, a chain $P'_h = \{x_i, a_h, b_h\}$ with $h(P'_h) = (2, 1)$, and a chain $P''_h = \{x_j, b_h, x_k\}$ with $h(P''_h) = (1, 2)$. Then, the path Π is a chain in \mathcal{P} with $h(\Pi) = (\nu + \alpha, 0)$. Eventually, for each occurrence of a variable x_i in a clause c_r , there is a chain from the variable leaf i to the leaf x_i in the clause gadget of c_r . Let us denote by P_i^r this chain. If the leaf x_i in the clause gadget of c_r is linked to a_r , then we have $h(P_i^r) = (|P_i^r| - 1, 1)$, else we have $h(P_i^r) = (|P_i^r| - 2, 2)$.

Now, the important point is that, because of all the chains of the form P_h , P'_h and P''_h , there are only 3 ways of coloring each clause gadget (see Figure 4.2: black vertices have color 1, white vertices have color 2).

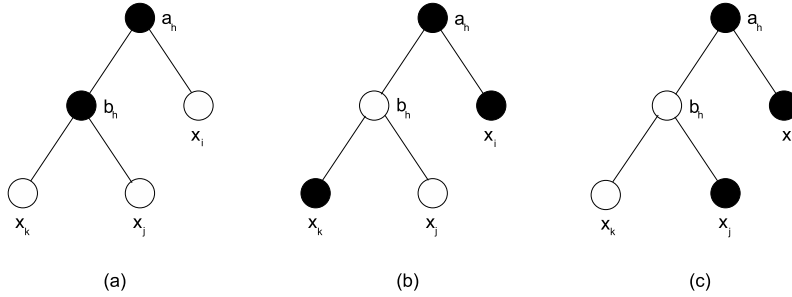


Figure 4.2: The 3 possible colorings for the clause gadget of $c_h = x_i \vee x_j \vee x_k$.

Moreover, because of all the chains of the form P_i^r , given one of the 3 possible colorings of the clause gadget of $c_h = x_i \vee x_j \vee x_k$, one and only one of the variable leaves labeled x_i , x_j and x_k has color 1: x_i in the coloring of Figure 4.2(a), x_j in the coloring of Figure 4.2(b), x_k in the coloring of Figure 4.2(c). Hence, given a solution for $\Lambda(G, 2, \mathcal{P}, H)$ on this instance, we can easily obtain a solution for the associated satisfiability instance, by assigning ‘true’ to variables whose variable leaves have color 1 and ‘false’ to the others. Conversely, given a truth assignment, assign color 1 to variable leaves associated with ‘true’ variables and color 2 to the others, and color each clause gadget with respect to the *only* variable equal to ‘true’ in the associated clause. It follows from the above discussion that we obtain a valid coloring. ■

In the above construction, by contracting Π into a single vertex v , and all the a_h into v (i.e., $a_1 = \dots = a_\alpha = v$) we obtain the following.

Theorem 4.6. $\Lambda(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete in trees of diameter at most 4 when $|P_i| \leq 4$ for each P_i in \mathcal{P} , $h_i^j \leq 3$ ($i \leq p$, $j = 1, 2$) and $|P_i \cap P_f| \leq 2$ for each P_i and P_f ($P_i \neq P_f$) in \mathcal{P} ($i, f \leq p$).

4.2.2 Polynomially solvable cases with $k = 2$

We recall that the basic image reconstruction problem in discrete tomography is polynomially solvable for $k = 2$ when the P'_i s are the rows and the columns of the associated grid graph G . Remember that in this special case we have $c(\mathcal{P}) = 2$.

More generally, we can state the following.

Theorem 4.7. $\Lambda(G, 2, \mathcal{P}, H)$ is polynomially solvable if $c(\mathcal{P}) = 2$.

Proof: We construct a multigraph G' as follows. Assign a vertex P_i to each chain P_i in \mathcal{P} . Each vertex of G , which is in P_i and in P_f is represented by an edge in G' between P_i and P_f . Each vertex, which is covered by a unique P_i is associated with an edge in G' between vertex P_i and a new vertex P'_i . So there is a one-to-one correspondence between the vertices of G and the edges of G' .

Then a solution, if there is one, will correspond to a subset F of edges of G' such that for each vertex P_i , F has h_i^1 edges adjacent to P_i (there is no restriction for the vertices P'_i).

In G' , the edges of F will give V^1 in G and the edges not in F will correspond to V^2 in G . There are polynomial algorithms (see [56]) to construct such subsets F if they exist or to decide that there is no solution. ■

One can derive the following from results in [39].

Theorem 4.8. $\Lambda(G, 2, \mathcal{P}, H)$ is polynomially solvable if $\text{Nest}(\mathcal{P}) = 2$.

Proof: Starting from the inclusion tree of each one of the two nested families covering \mathcal{P} , one can build a network flow model where a compatible integral flow will define the subset $V^1 \subseteq V$ and $V^2 = V - V^1$ will be obtained immediately as shown in [39].

Assume \mathcal{P} can be decomposed into nested subfamilies A and B . We represent both families by the inclusion tree of their subsets P_i . A source a (resp. a sink b) is linked to all maximal (inclusionwise) subsets of A (resp. B). We link each $l \in V$ to the unique minimal subset A_r of A (resp. B_s of B) which contains l by an arc (A_r, l) (resp. (l, B_s)). The network is obtained by orienting all remaining edges from a to b . The arc entering (resp. leaving) each P_i in A (resp. B) has a capacity and a lower bound of flow equal to h_i^1 . The arcs adjacent to the vertices corresponding to the elements of V have capacity 1 and a lower bound of flow equal to 0.

In Figure 4.3 an example is given for a set $V = \{1, 2, \dots, 7\}$ and a family \mathcal{P} with $\text{Nest}(\mathcal{P}) = 2$. Here $A = (\{1, 2\}, \{3, 4, 5\}, \{6, 7\})$ and $B = (\{1, 3, 6\}, \{2, 4\}, \{5, 7\}, \{1, 3, 5, 6, 7\})$. The values h_i^1 are shown in brackets.

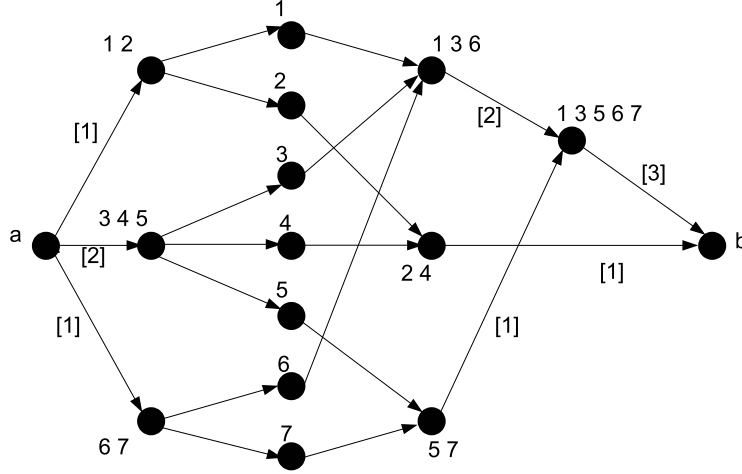


Figure 4.3: The network associated with a family \mathcal{P} with $\text{Nest}(\mathcal{P}) = 2$.

There is a one-to-one correspondence between the feasible integral flows from a to b and the subset V^1 of vertices in a coloring (V^1, V^2) satisfying the requirements. ■

Theorem 4.9. *Let G be an arbitrary graph and \mathcal{P} a family of chains P_i such that any P_i has at most two vertices belonging to some other chains of \mathcal{P} . Then $\Lambda(G, 2, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: We shall transform the problem into a *2SAT* problem which is known to be polynomially solvable [3].

We associate a binary variable x with every vertex of G which belongs to at least two chains P_i . Notice that we may assume that $\min\{h_i^1, h_i^2\} \geq 1$, $i \leq p$, otherwise there is only one color occurring in P_i and the problem can be reduced. We first remove all vertices which belong to exactly one P_i (these will be considered later). Now each P_i contains one or two vertices. For each P_i which has exactly two vertices, say x and y , which belong to other chains, we write a clause c_i as follows.

If $h_i^1 = 2, h_i^2 = 1$, we set $c_i = x \vee y$ (this means that at least one of the vertices x, y must have color 1) and if $h_i^1 = 1, h_i^2 = 2$, we set $c_i = \bar{x} \vee \bar{y}$ (at least one of x, y must have color 2). If $\min\{h_i^1, h_i^2\} \geq 2$, we do nothing (since x and y can get any color). Finally when $h_i^1 = h_i^2 = 1$, we introduce a constraint $x = \bar{y}$ (because x and y must get different colors). For any P_i which has exactly one vertex belonging to more than one chain in \mathcal{P} , we do nothing since by assumption ($\min\{h_i^1, h_i^2\} \geq 1$) this vertex can have any color. We define $\mathcal{C} = \bigwedge_{i=1}^q c_i$ and using the equality constraints $x = \bar{y}$ we may substitute variable \bar{y} to variable x . We are left with a *2SAT* instance. It has a solution if and only if $\Lambda(G, 2, \mathcal{P}, H)$

has a solution.

From a solution of $2SAT$, we derive a partition V^1, V^2 of the vertices associated with the binary variables. The bicoloring V^1, V^2 of the vertices of G belonging to more than one chain of \mathcal{P} is given by $V^1 = \{v \mid v \text{ is true}\}$, $V^2 = \{v \mid v \text{ is false}\}$. For each P_i it is possible to assign color 1 or 2 to the uncolored yet vertices so that the number of occurrences of color j is h_i^j (for $j = 1, 2$). This will provide the required coloring of G .

Conversely if $\Lambda(G, 2, \mathcal{P}, H)$ has a solution, then by setting x to ‘true’ (resp. x ‘false’) for all variables corresponding to the vertices x which are in more than one chain and have color 1 (resp. color 2), we will satisfy all clauses in \mathcal{C} (as well as the equality constraints). ■

Theorem 4.10. *If $G = (V, U)$ is a directed tree and each $P_i \in \mathcal{P}$ is a directed path, then $\Lambda(G, 2, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: Notice that the incidence matrix (paths \times vertices) of such a graph is totally unimodular. So if we write the system $Ax = b$, $\mathbf{0} \leq x \leq \mathbf{1}$ where $a_{iv} = 1$ if path P_i contains vertex v (or $a_{iv} = 0$ else) and $b_i = h_i^1$, then we may check in polynomial time with a linear programming solver whether the system has a solution; if it is the case there is an integral solution (since A is totally unimodular) which gives V^1 , and $V^2 = V - V^1$ which form a partition of V satisfying all requirements. ■

4.2.3 The case $k \geq 3$

Let us first consider the special case where all P_i 's have size $|P_i| \leq 2$.

Theorem 4.11. *For any graph G and any \mathcal{P} such that every $|P_i| \leq 2$, $\Lambda(G, k, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: Consider $\Lambda(G, k, \mathcal{P}, H)$. Eliminate all P_i 's such that $h_i^j = 2$ for some color $j \leq k$ (these have a unique coloring) and apply the reductions implied by these eliminations. We also apply the reductions due to chains P_i with $|P_i| = 1$.

Consider a pair P_i, P_f with $|P_i \cap P_f| = 1$. Let Π_l be the set of colors j with $h_i^j > 0$. If $\Pi_i \cap \Pi_f = \emptyset$, there is no solution; if $|\Pi_i \cap \Pi_f| = 1$, then assign this color to the vertex in $P_i \cap P_f$ and the rest of P_i, P_f is also determined. We apply these reductions until either we get a contradiction or we have a collection of connected components C_1, \dots, C_r where in each connected component all P_i 's have the same set Π_i of possible colors (remember that $|P_i| = 2$ and $|\Pi_i| = 2$). Then our problem has a solution if and only if every connected component is bipartite. ■

For the case where the number of colors is $k = 3$, we have the following.

Theorem 4.12. *$\Lambda(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete when $|P_i| = 3, h_i^j = 1$ for $i = 1, \dots, p, j = 1, 2, 3$ and $c(\mathcal{P}) = 2$.*

Proof: We use a transformation from edge 3-coloring of a 3-regular graph G' . This problem is known to be \mathcal{NP} -complete [43].

We will construct a graph G and a family \mathcal{P} of chains in G . We will associate a chain P_i in G with each vertex w_i of G' ; each edge $[w_i, w_f]$ of G' is associated with a vertex $v_{if} \equiv v_{fi}$ of $V(G)$. P_i will be a chain in G containing the three vertices corresponding to the three edges adjacent to w_i in G' . If in G' vertex w_i is adjacent to w_r, w_s and w_t ($r < s < t$), then in G , $P_i = \{v_{ir}, v_{is}, v_{it}\}$, and the corresponding chain will be formed by edges $[v_{ir}, v_{is}], [v_{is}, v_{it}]$.

We set $h_i^j = 1$ for $i = 1, \dots, p$ and $j = 1, 2, 3$. Then there is an edge 3-coloring of G' if and only if there is a partition V^1, V^2, V^3 of $V(G)$ such that for each $P_i, |P_i \cap V^j| = 1 = h_i^j$ for any i, j . ■

Theorem 4.12 is best possible since from Theorem 4.11 the problem is easy when $|P_i| \leq 2$ for all $i \leq p$.

Remark 4.13. According to Brooks' theorem (see [9]), the chromatic number $\chi(G)$ of a 3-regular connected graph G is 3 unless G is either a clique on four vertices (in which case $\chi(G) = 4$) or a bipartite graph (in which case $\chi(G) = 2$).

Since edge 3-coloring is \mathcal{NP} -complete in 3-regular graphs [43], we can state: edge 3-coloring in a 3-regular graph G is \mathcal{NP} -complete even if $\chi(G) = 3$.

Conversely, note that if a connected graph G is edge 3-colorable then $\Delta(G) \leq 3$ and thus either G is a clique on four vertices or $\chi(G) \leq 3$.

4.2.4 The case where G is a chain or a tree and $k > 2$

We will now consider $\Lambda(G, k, \mathcal{P}, H)$ where G is a tree, each P_i is a chain of G and furthermore for any two chains P_i, P_f in \mathcal{P} we have $|P_i \cap P_f| \leq 1$. In such a case we have the following.

Lemma 4.14. If G is a tree and if the family \mathcal{P} of chains of G satisfies $|P_i \cap P_f| \leq 1$ for all $i, f \leq p$, then there is an order (which we call canonical order) of chains such that for any $q > 1$

$$|P_q \cap (\bigcup_{i=1}^{q-1} P_i)| \leq 1$$

Proof: Notice first that we can assume $|P_i| \geq 2$ for each $i \leq p$. This implies that we cannot have $P_i \subset P_f$, for any $i, f \leq p$ ($i \neq f$). Now G has a pendent vertex contained in exactly one chain P_i of \mathcal{P} . This chain will be called P_1 ; we remove it from \mathcal{P} as well as all vertices belonging to P_1 only. Now we can find another pendent vertex of the remaining tree G' and this determines P_2 . We will thus find a numbering of the chains of \mathcal{P} which satisfies the requirements. ■

We will describe below an algorithm for solving $\Lambda(G, k, \mathcal{P}, H)$ in a tree $G = T$; in this procedure (called FFC) we will have to determine for each vertex of T the 'forced' colors as well as the 'forbidden' colors; such a procedure will also be able to detect contradictions in

the data which imply that no solution exists. A color c is said to be *forced* (resp. *forbidden*) for a vertex v if there exists no feasible solution where v has a color $c' \neq c$ (resp. where v has color c).

The procedure FFC which makes a repeated use of a maximum flow in a bipartite graph can be sketched as follows.

Procedure FFC (Forced and Forbidden Colors): Let us consider a chain P_i and let us denote by x_1, \dots, x_ν the vertices in P_i . Let Π_i be the set of colors required in P_i , i.e., $\Pi_i = \{j \mid h_i^j > 0\}$. For each vertex x_l , $l = 1, \dots, \nu$, π_l denotes the set of possible colors for x_l , i.e. $\pi_l = \bigcap_{i \mid x_l \in P_i} \Pi_i$.

We construct the following bipartite graph $G = (X, Y, E)$ with $X = \{x_1, \dots, x_\nu\}$, $Y = \Pi_i$, and $[x_l, j] \in E$ if $j \in \pi_l$; the capacity of $[x_l, j]$ is equal to 1. To get a network N , we add a source s with an arc of capacity 1 from s to each vertex in X and a sink t with an arc from each vertex j in Y to t ; the capacity of (j, t) is equal to h_i^j for all $j \in Y$. Any integral flow from s to t saturating the arcs out of s gives a possible coloring of the vertices in P_i . To any edge $[x_l, j] \in E$ which is saturated in every maximum flow corresponds a forced color j for x_l . To any edge $[x_l, j] \in E$ with a flow equal to 0 in every maximum flow corresponds a color j forbidden for x_l .

Note that it is easy to determine all the edges saturated (resp. with no flow) in every maximum flow. For each edge $[x_l, j]$ in E , suppress $[x_l, j]$ (resp. force a flow from s to t through $[x_l, j]$) and compute a new maximum flow in the obtained network. If the value of this flow is lower than the original maximum flow, then $[x_l, j]$ is saturated (resp. with no flow) in every maximum flow.

Procedure FFC either finds the forbidden colors or a forced color for a vertex v or concludes that there is no more forbidden color nor forced color. If the set π_v of possible colors for v is $\pi_v = \{1, \dots, k\}$ initially for each vertex v , we notice that finding a forced color c for v reduces π_v to a set $\pi_v = \{c\}$ and finding the forbidden colors c_{i_1}, \dots, c_{i_q} for v replaces π_v by $\pi_v = \pi_v - \{c_{i_1}, \dots, c_{i_q}\}$.

Since we will apply FFC as long as forbidden or forced colors can be found, it will be called at most $|V|k$ times.

Clearly we remove all vertices which have a forced color and we update the values h_i^j accordingly as well as the sets Π_i .

At the end of the repeated applications of FFC we will either have discovered a contradiction ($\pi_v = \emptyset$ for some vertex v) or obtained for each vertex v a set π_v with $|\pi_v| \geq 2$.

Theorem 4.15. *If G is a tree and \mathcal{P} a family of chains of G satisfying $|P_i \cap P_f| \leq 1$ for any $i, f \leq p$ ($i \neq f$), then $\Lambda(G, k, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: We start by applying the FFC procedure; it may happen that one has to remove some vertices with forced colors; in such a case we get a forest and we apply the procedure

on each connected component separately.

W.l.o.g. we consider a tree G and we construct a canonical order P_1, \dots, P_p of the chains of \mathcal{P} . Since we apply procedure FFC until there are no more forced colors and neither forbidden colors, we have the following.

Fact 4.16. *If in a chain P_i a single arbitrary vertex v has been given a possible color $c \in \pi_v$, there exists an assignment of possible colors $c(w) \in \pi_w$ to all remaining vertices w of P_i such that P_i has exactly h_i^j vertices of color j ($1 \leq j \leq k$).*

It is then possible to color the vertices of G by considering the chains P_1, \dots, P_p in the canonical order (starting from any vertex of P_1). Clearly we will be able to extend the coloring to all vertices of G since, having colored the vertices of P_1, \dots, P_i , the chain P_{i+1} has exactly one vertex v which is already colored (with a color in π_v).

The whole procedure is polynomial:

FFC consists of applying for each chain P_i a maximum flow algorithm in a bipartite network with $|P_i|$ vertices on the left and k vertices on the right. To find the forbidden colors and the forced colors, we have to find at most $|P_i|k$ times an augmenting chain (this takes $O(|P_i|k)$ time); globally we have a complexity $O((|P_i|k)^2)$ for getting the forbidden colors and the forced colors. For a maximum flow we have $O((|P_i| + k)^3)$ (see [1]). Hence an application of FFC has a complexity $O((|P_i| + k)^3 + (|P_i|k)^2)$. Since we apply FFC at most $|V|k$ times, we have $O(((|P_i| + k)^3 + (|P_i|k)^2)|V|k)$ and since $|P_i| \leq |V|$ we finally have $O(((|V| + k)^3 + (|V|k)^2)|V|k)$. ■

Proposition 4.17. *If G is a cycle and if the family \mathcal{P} is such that $|P_i \cap P_f| \leq 1$ for any $i, f \leq p$ with $i \neq f$, then $\Lambda(G, k, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: We take a consecutive numbering of the chains P_i , as in the case where G is a tree so that $|P_i \cap P_{i+1}| = 1$ for all $i \leq p - 1$ and in addition $|P_p \cap P_1| = 1$; let $v_0 \in P_p \cap P_1$.

We simply consider the following problems O_j (for $j = 1, \dots, k$): find a feasible coloring such that v_0 has color j .

This amounts to removing v_0 and updating the h_i^j accordingly; this is simply $\Lambda(G - v_0, k, \mathcal{P}', H')$ where $G - v_0$ is a chain. ■

More generally if G is a *cactus*, i.e., a connected graph where any two cycles have at most one common vertex, then we can proceed as for a tree in the following special case. Let us assume that each P_i belongs to exactly one cycle (or to a chain not contained in a cycle). Each cycle C has some vertices which may belong to other cycles or to external chains; we shall assume that all these vertices are necessarily endpoints of chains P_i .

It is not difficult to see that we can number the P_i 's in \mathcal{P} in such a way that for all $f \leq p$ $|P_f \cap (\bigcup_{i=1}^{f-1} P_i)| \leq 1$ (except for the last P_i 's which 'close' a cycle in G).

We can work separately on each cycle C and determine the possible colors for the last vertex,

i.e., the vertex connecting C to some cycle or some external chain covered by chains P_i with smaller indices.

We proceed as in the case of trees by applying an FFC procedure first and then, in case no contradiction has occurred, we will be in the situation where we have either a single P_i (contained in an external chain) to color where exactly one vertex is already colored or we reach a cycle (with exactly one vertex already colored). In the first case we proceed as before and in the second one, we have that the cycle can be colored by extending the coloring from the vertex which has been colored and we continue. This will finally color the whole graph. As in the case of trees, the procedure will give a feasible coloring or exhibit a contradiction.

4.3 Proper colorings

Having discussed $\Lambda(G, k, \mathcal{P}, H)$ we shall examine the case where the k -partition is a proper k -coloring ($\Lambda^*(G, k, \mathcal{P}, H)$).

Here we shall assume that for every edge $e = [x, y]$ in G , there is a chain μ_i which uses e ; this implies in particular $x, y \in P_i$. This assumption is not restrictive: let $e = [x, y]$ be an edge which is not covered by any μ_i in the collection defined in $\Lambda^*(G, k, \mathcal{P}, H)$. We replace e by a chain $\mu_e = \{x_e^1 = x, u_e^1, x_e^2, u_e^2, \dots, x_e^{k-1}, u_e^{k-1}, x_e^k = y\}$ where x_e^2, \dots, x_e^{k-1} are new vertices and u_e^1, \dots, u_e^{k-1} are new edges; we set $P_e = \{x_e^1, \dots, x_e^k\}$ and $h_e^j = 1$ for $j = 1, \dots, k$.

There is a proper k -coloring of the resulting graph G^* which is solution of $\Lambda^*(G^*, k, \mathcal{P}, H)$ if there is a proper k -coloring which is solution of $\Lambda^*(G, k, \mathcal{P}, H)$, because x and y will necessarily get different colors in any feasible coloring of G^* .

4.3.1 Solvable cases of proper colorings

Let us now consider some cases for which polynomial time algorithms can be found.

Fact 4.18. $\Lambda^*(G, 2, \mathcal{P}, H)$ is polynomially solvable.

Justification: Notice that in each P_i with odd $|P_i|$, the vertices have necessarily forced colors. So we can assume that there are only chains of even length and each vertex may be colored with color 1 or 2. The problem then consists in verifying whether the graph is bipartite or not, which can be done in polynomial time.

We obtain from Theorem 4.15 and its proof.

Corollary 4.19. $\Lambda^*(G, k, \mathcal{P}, H)$ can be solved in polynomial time if G is a tree, \mathcal{P} is such that $|P_i \cap P_f| \leq 1$ for $i, f \leq p$ ($i \neq f$) and $h_i^j \leq 1$ for all $i \leq p, j \leq k$.

From now on we will have to consider repeatedly proper k -colorings of chains P_i of G (with possibly $k > 2$ and with h_i^j occurrences of color j in chain P_i). So we will start by stating some elementary properties of such colorings.

We recall that a color j is saturating in a chain P if $h^j = \lceil \frac{|P|}{2} \rceil$. The set of colors j such that $h^j > 0$ will be denoted by Π .

Remark 4.20. *If P is an odd chain with a saturating color a , then a occurs necessarily at both endpoints of P in any coloring.*

Remark 4.21. *If P is an even chain with a saturating color a , then a occurs necessarily at least at one endpoint of P in any coloring.*

Lemma 4.22. *Let P be a chain to be colored and assume there is no saturating color in Π . For any two colors e, d in Π , one can find a proper k -coloring of P where e and d occur at the endpoints of P . In case $h^d \geq 2$, we can have a coloring with d occurring at both endpoints.*

Proof: Let $P = \{1, 2, \dots, n\}$ and let d and e be the colors which have to occur at the ends. Assume first that n is even. Start from the left, assigning h^d times color d to vertices $1, 3, \dots, 2h^d - 1$ and from the right, assign h^e times color e to vertices $n, n - 2, \dots, n - 2(h^e - 1)$. It remains $\max\{0, n - 2h^e - 2h^d + 2\}$ adjacent vertices in the center. We can find $\max\{0, \frac{n}{2} - h^e - h^d + 1\}$ nonadjacent vertices among them. Together with the vertices $2, 4, \dots, 2h^d - 2$ and $n - 1, n - 3, n - 2h^e + 3$, this gives $\frac{n}{2} - 1$ non adjacent vertices. If n is odd, we choose a color $f \neq d, e$ (which exists since there is no saturating color). We color vertex n with e and we decrease h^e by one. Then we apply the previous coloring, with color f replacing color e , to $P' = P - \{n\}$; this will give a proper coloring of P since vertex $n - 1$ has color f and vertex n has color e .

Finally we start by coloring the non adjacent vertices with the remaining colors. If $h^e + h^d - 1 \geq \frac{n}{2}$, then all uncolored vertices are nonadjacent and the coloring can be completed. In the other case ($h^e + h^d - 1 < \frac{n}{2}$), we have an interval I of $n - 2h^e - 2h^d + 2$ consecutive uncolored vertices in the center. We color the remaining vertices in the order $2h^d, 2h^d + 2, \dots, n - 2h^e, n - 2h^e + 3, \dots, n - 1, 2, 4, \dots, 2h^d - 2, 2h^d + 1, 2h^d + 3, \dots, n - 2h^e + 1$ exhausting one color before taking the next one. Since there is no saturating color we will get a proper coloring of the chain.

To obtain a coloring with d occurring on 1 and n , consider $P' = P - \{n\}$ and $(h^d)' = h^d - 1$. Apply the coloring algorithm to P' with colors d and e . Clearly vertex $n - 1$ will not have color d and we can color vertex n with d to get the required proper coloring of P . ■

Lemma 4.23. *If P is an even chain with exactly one saturating color a , one can choose any color b and construct a coloring of P such that a and b are occurring at the endpoints.*

Proof: Assume first $b \neq a$. Color the vertices $1, 3, 5, \dots, |P| - 1$ with color a and color the vertices $|P|, |P| - 2, \dots, 2$ with the remaining colors starting with color b .

If $b = a$, then color a occurs at both ends: we color vertices $1, 3, 5, \dots, |P| - 3$ and $|P|$ with color a . Since there are no other saturating colors, we can color the vertices $|P| - 2, |P| - 4, \dots, 2, |P| - 1$ with the remaining colors and no conflict will occur. ■

We shall say that the singletons P_i in \mathcal{P} have the *CS* property (Consecutive Singletons) if

the following holds: if a singleton P_i is an intermediate vertex of some $P_c = \{x_c^1, x_c^2, \dots, x_c^r = P_i, x_c^{r+1}, \dots, x_c^s\}$, then either x_c^1, \dots, x_c^{r-1} or x_c^{r+1}, \dots, x_c^s are also singletons in \mathcal{P} .

Proposition 4.24. *Let G be a chain. For any $\Lambda^*(G, k, \mathcal{P}, H)$ with $|P_i \cap P_f| \leq 1 \ \forall i \neq f (i, f \leq p)$ and where all singletons P_i have the CS property, there is an equivalent problem $\Lambda^*(G^*, k, \mathcal{P}^*, H^*)$ where the family \mathcal{P}^* satisfies:*

(a) $|P_i^*| \geq 2$;

(b) $|P_i^* \cap P_{i+1}^*| \leq 1 \ (i < p^*)$ and $P_i^* \cap P_f^* = \emptyset \ (i \notin \{f-1, f, f+1\})$.

Here ‘equivalent’ means that one problem has a solution if and only if the other one has a solution.

Proof: Assuming that the vertices are given in increasing order of numbering along the chain, we can say that a chain P_i starts at some vertex x_d (or ends at some vertex x_e) if d is the smallest (e is the largest) index in P_i .

Now consider a chain $P_c = \{x_c^1, \dots, x_c^r, x_c^{r+1}, \dots, x_c^s\}$ where x_c^1, \dots, x_c^r are singletons P_c^1, \dots, P_c^r in \mathcal{P} . We remove x_c^1, \dots, x_c^r and replace P_c by $P_c^* = \{x_c^{r+1}, y^1, \dots, y^{k-2}\}$ with $h(P_c^*) = (1, \dots, 1, 0, 1, \dots, 1)$ where the missing color is the color of x_c^r and y^1, \dots, y^{k-2} are new vertices. We also introduce $P_c^{**} = \{x_c^{r+1}, \dots, x_c^s\}$ with updated values of h_i^j according to the colors already assigned to x_c^1, \dots, x_c^r . Similarly, if there is a chain $P_d = \{x_d^1, \dots, x_d^t = x_c^1\}$ ending at vertex x_c^1 , we replace it by a chain $P_d^* = P_d - x_c^1 = \{x_d^1, \dots, x_d^{t-1}\}$ and introduce $P_d' = \{x_d^{t-1}, z^1, \dots, z^{k-2}\}$ with $h(P_d') = (1, \dots, 1, 0, 1, \dots, 1)$ where the missing color is the color of x_c^1 . We update the values h_i^j accordingly. Then we have an equivalent problem since x_d^{t-1} will not get the color of x_c^1 and x_c^{r+1} will not get the color of x_c^r . So we have cut the problem into two subchains and singletons in P_c have been removed. By repeating this we get an equivalent problem with all P_i 's verifying $|P_i| \geq 2$. ■

Theorem 4.25. *If G is a chain where all the singletons P_i have the CS property and \mathcal{P} is such that $|P_i \cap P_f| \leq 1 \ \forall i \neq f \ (i, f \leq p)$, then $\Lambda^*(G, k, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof: As already remarked, we can assume that $k \geq 3$.

W.l.o.g. we can assume that \mathcal{P} has the properties (a) and (b) given in Proposition 4.24. Consider now the problem $\Lambda^*(G, k, \mathcal{P}, H)$. To solve it we use a procedure similar to the one used for $\Lambda(G, k, \mathcal{P}, H)$. If any contradiction occurs during the following forced assignments then there is no solution.

- whenever a vertex $v \in P_i \cap P_{i+1}$ is assigned some color j we update the parameters as follows: $h_i^j \leftarrow h_i^j - 1$; $h_{i+1}^j \leftarrow h_{i+1}^j - 1$; if $h_i^j = 0$, then set $\Pi_i \leftarrow \Pi_i - \{j\}$; if $h_{i+1}^{j+1} = 0$, then set $\Pi_{i+1} \leftarrow \Pi_{i+1} - \{j\}$;
- if there exists $1 \leq i < p$ such that $\Pi_i \cap \Pi_{i+1} = \emptyset$, then there is no solution;

- if there exists $1 \leq i < p$ such that $|II_i \cap II_{i+1}| = 1$, then color $P_i \cap P_{i+1}$ with the common color;
- for each odd P_i with a saturating color, say j , assign color j to both endpoints of P_i ;
- for each even P_i with a saturating color, say j , j must be assigned to one of the endpoints of P_i . For $1 < i < p$,

if $j \notin II_{i-1} \cup II_{i+1}$, then there is no solution;

if $j \notin II_{i-1}$, then assign color j to $P_i \cap P_{i+1}$;

if $j \notin II_{i+1}$, then assign color j to $P_i \cap P_{i-1}$.

For any colored vertex, propagate the possible implications of this coloring to the previous and next intersections in the following way; if any contradiction occurs, there is no solution.

Assume that $v \in P_i \cap P_{i+1}$ has been colored with j :

- if $l \neq j$ is a saturating color of P_i (resp. P_{i+1}), then color the left (resp. right) endpoint of P_i (resp. P_{i+1}) with l ;
- if $|II_i \cap II_{i-1}| = 1$ ($i > 1$) (resp. $|II_{i+1} \cap II_{i+2}| = 1$ ($i < p - 2$)), assign the unique color l such that $h_i^l \geq 1$ and $h_{i-1}^l \geq 1$ (resp. $h_{i+1}^l \geq 1$ and $h_{i+2}^l \geq 1$) to $P_i \cap P_{i-1}$ (resp. $P_{i+1} \cap P_{i+2}$).

At this step, if no contradiction occurred, we have a set of colored vertices located at intersections of chains P_i . In addition, any pair $\{P_i, P_{i+1}\}$ ($i < p$) such that $P_i \cap P_{i+1}$ is uncolored verifies $|II_i \cap II_{i+1}| \geq 2$ and if j is a color saturating P_i then $j \in II_{i-1} \cap II_i \cap II_{i+1}$, $1 < i < p$.

Moreover, if one endpoint of P_i ($1 \leq i \leq p$) is already colored, any color remaining in II_i is compatible with it and can be used to color the other endpoint; if P_i has a saturating color it is the one already assigned.

The problem has a solution which can be obtained in two more steps.

- (A) First we assign a color to all uncolored intersection $P_i \cap P_{i+1}$ ($i < p$), in the following way. Let $P_i \cap P_{i+1}$ be the first uncolored intersection in G ; color $P_i \cap P_{i+1}$ with any color $j \in II_i \cap II_{i+1}$. If $i = 1$, color the first endpoint of P_i with any allowed color. If $P_{i+1} \cap P_{i+2}$ is uncolored ($i + 1 < p$) then there is at least one color different from j in $II_{i+1} \cap II_{i+2}$; if there is a saturating color l in P_{i+1} , and if $l \neq j$ then assign color l to $P_{i+1} \cap P_{i+2}$ (we are sure that $l \in II_{i+1} \cap II_{i+2}$) otherwise choose any color in $II_{i+1} \cap II_{i+2}$. Propagate the implications of each coloring until we reach a vertex already colored. Then search for the following uncolored intersection and continue the process until the end of G .

(B) Clearly the partial coloring obtained so far is such that for every chain the saturating colors are assigned to endpoints in such a way that according to Lemmas 4.22 and 4.23, the coloring can be extended to all yet uncolored vertices. ■

Remark 4.26. *One should mention that Theorem 4.25 can be extended to trees where \mathcal{P} is such that in every P_i only the ‘first’ and ‘last’ vertices may belong to another P_j .*

Remark 4.27. *$\Lambda^*(G, k, \mathcal{P}, H)$ can be solved in polynomial time if $|P_i| = 2$ for all $i = 1, \dots, p$. Since $|P_i| = 2$ for each i , each edge is a P_i and there are exactly two possible colorings for each P_i . We take the first coloring of P_1 ; we propagate this coloring and if we obtain a proper coloring of G , we are done. Else we have a conflict; we then reverse the coloring of P_1 and propagate this coloring as before and we will find a coloring of G or a conflict. In the last case, there is no solution.*

4.3.2 Difficult cases of proper colorings

Theorem 4.28. *$\Lambda^*(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete in trees with maximum degree 3.*

Proof: We use the construction in the proof of Theorem 4.5 and introduce a new vertex on each edge of the tree; we force these new vertices to have color 3. ■

Theorem 4.29. *$\Lambda^*(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete even if G is planar bipartite, $|P_i \cap P_f| \leq 1$ ($i, f \leq p, i \neq f$), $|P_i| \leq 3$ ($i \leq p$) and $h_i^j \leq 1, i = 1, \dots, p, j = 1, 2, 3$.*

Proof: We use a transformation from the \mathcal{NP} -complete problem *PrExt* which is defined as follows.

Instance: A positive integer q and a graph G in which some vertices are precolored using at most q colors.

Question: Can the precoloring of G be extended to a proper coloring of G using at most q colors?

This problem is proven to be \mathcal{NP} -complete even if $q = 3$ and G is planar bipartite (see [51]).

Consider a planar bipartite graph $G = (X, Y, E)$. Suppose that some of its vertices are precolored using colors 1, 2 and 3. For each precolored vertex x , we set $P_x = \{x\}$ and $h(P_x) = (1, 0, 0)$ if x has color 1, $h(P_x) = (0, 1, 0)$ if x has color 2 and $h(P_x) = (0, 0, 1)$ if x has color 3. For each edge $e = [x, y]$ in G , we add a new vertex z_e and a new edge $[x, z_e]$. We set $P_e = \{x, y, z_e\}$ and $h(P_e) = (1, 1, 1)$.

Clearly our new graph G' is still planar bipartite. Furthermore $|P_i \cap P_f| \leq 1$ ($i, f \leq p, i \neq f$), $|P_i| \leq 3$ ($i \leq p$) and $h_i^j \leq 1, i = 1, \dots, p, j = 1, 2, 3$.

It is easy to see that *PrExt* has a solution in G if and only if $\Lambda^*(G', 3, \mathcal{P}, H)$ has a solution in G' . ■

4.4 Edge colorings

We now consider edge colorings instead of vertex colorings; we may in a similar way define problem $\Psi(G, k, \mathcal{P}, H)$ where \mathcal{P} is a collection of p subsets P_i of edges of G and we want to find a k -partition E^1, E^2, \dots, E^k of E such that

$$|P_i \cap E^j| = h_i^j \text{ for all } i \leq p \text{ and all } j \leq k. \quad (4.2)$$

If we want to find a proper edge k -coloring then the problem will be denoted by $\Psi^*(G, k, \mathcal{P}, H)$.

In general the subsets P_i of edges will be chains (open or closed). $|P_i|$ will be the number of edges in chain P_i .

Clearly problems Ψ and Ψ^* in a graph G are equivalent to problems Λ and Λ^* in $L(G)$ where $L(G)$ is the line graph of G (edges of G become vertices of $L(G)$).

It follows that when G itself is a chain, then $L(G)$ is also a chain and the results for Λ and Λ^* also apply to the edge coloring case.

4.4.1 Arbitrary colorings

In this situation every edge e which is not included in some P_i may be removed from G . So we can assume w.l.o.g. that every e is in some P_i of \mathcal{P} .

Theorem 4.30. *$\Psi(G, k, \mathcal{P}, H)$ can be solved in polynomial time if $|P_i| \leq 2$ for each chain $P_i \in \mathcal{P}$.*

Proof: This follows directly from the proof of Theorem 4.11. After reduction we transform the graph as follows: each edge becomes a vertex and we link two vertices if there is a P_i containing the corresponding edges. The problem has a solution if and only if there is no odd cycle in this graph. ■

Theorem 4.31. *$\Psi(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete even if G is a tree T with maximum degree 3 and the P_i 's are chains or bundles.*

Proof: We use the same reduction from ONE-IN-THREE 3SAT as in Theorem 4.5.

We have to color edges instead of vertices; the leaf variables now correspond to leaf edges and for each clause c_h we now have for the sets P'_h bundles of edges y, z and x (see Figure 4.4(a)) if the clause is given by $c_h = x_i \vee x_j \vee x_k$. We set $h(P'_h) = (2, 1)$. The set P''_h is now the bundle y, u, t with $h(P''_h) = (1, 2)$ and the set $P_h = \{x\}$ with $h(P_h) = (1, 0)$. The other chains P_i^r are defined similarly. ■

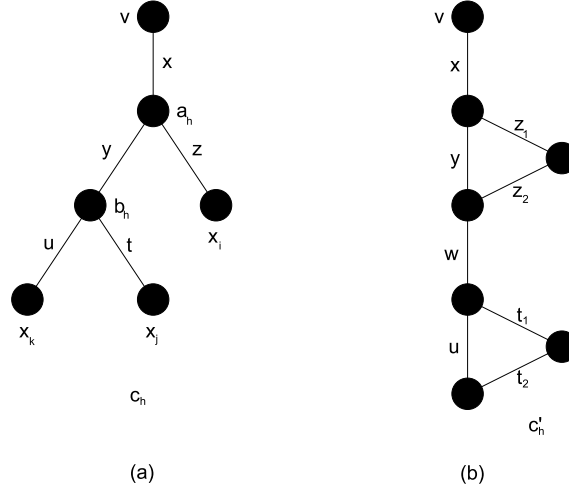


Figure 4.4: Transformation of bundle constraints for a tree into chain constraints in a cactus.

If we require all P_i 's to be exclusively chains in G (but not bundles) we can derive the following for a special cactus in which no two cycles have a common vertex (see [10] for additional properties of cacti).

Theorem 4.32. $\Psi(G, 2, \mathcal{P}, H)$ is \mathcal{NP} -complete even if G is a triangulated cactus (in which no two cycles have a common vertex) with maximum degree 3 and where all the P_i 's are chains.

Proof: We just have to show how the bundle requirements can be transformed into constraints related to chains.

We transform the clause gadget c_h as shown in Figure 4.4(b). The cactus obtained in this way is triangulated (its cycles are triangles).

The bundle $P' = \{x, y, z\}$ with $h(P') = (2, 1)$ becomes chains $P'_* = \{x, y, z_2, z_1\}$, $P'_{**} = \{z_2\}$ with $h(P'_*) = (2, 2)$, $h(P'_{**}) = (0, 1)$.

The bundle $P'' = \{y, u, t\}$ with $h(P'') = (1, 2)$ becomes chains $P''_* = \{y, w, u, t_2, t_1\}$, $P''_{**} = \{w, t_1\}$ with $h(P''_*) = (1, 4)$ and $h(P''_{**}) = (0, 2)$.

Finally the P_i 's using chains between v and edges u and t can also be replaced by chains in the new gadget c'_h with appropriate modifications of the values h_i^j . ■

4.4.2 Proper colorings

Theorem 4.33. $\Psi^*(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete when G is 3-regular, \mathcal{P} is a collection of vertex disjoint triangles P_i considered as sets of edges (i.e. $|P_i| = 3, \forall i = 1, \dots, p$, $P_i \cap P_f = \emptyset$ for all $i, f, i \neq f$) and $h_i^j = 1, \forall i = 1, \dots, p, \forall j = 1, 2, 3$.

Proof: We use a transformation from edge 3-coloring of a 3-regular graph G' . This problem is known to be \mathcal{NP} -complete [43].

For each vertex i adjacent to vertices f, l and p , we introduce in G the vertices v_{if}, v_{il} and v_{ip} . These three vertices are pairwise linked forming a triangle which will correspond to P_i . Thus G will have $3|V|$ vertices. For each edge $[i, f]$ in G' we introduce an edge $[v_{if}, v_{fi}]$ in G .

We take $p = |V(G')|$ and $\mathcal{P} = (P_1, \dots, P_p)$ with $h_i^j = 1$ for $i = 1, \dots, p$ and $j = 1, 2, 3$. Notice that the P_i 's form closed chains.

There is an edge 3-coloring of G' if and only if there is an edge 3-coloring of G . The edges of G are colored as follows:

- (1) for each edge $[i, f]$ of G' with color k , the corresponding edge $[v_{if}, v_{fi}]$ in G has color k ;
- (2) the three edges forming a triangle P_i can be colored with three colors by extending the coloring obtained after the previous stage.

Finally, note that any edge 3-coloring of G will satisfy the requirements on the sets P_i . ■

Theorem 4.34. $\Psi^*(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete when G is a bipartite 3-regular graph and \mathcal{P} is a family of chains P_i of length two which are pairwise non adjacent.

Proof: Let us call SIM (for simultaneity requirements) the following problem. We are given a 3-regular bipartite simple graph G^* with two subsets \mathcal{S}_1 and \mathcal{S}_2 of edges such that $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and the edges of \mathcal{S}_i are pairwise non adjacent for $i = 1, 2$.

Does there exist an edge 3-coloring (M_1, M_2, M_3) of G^* such that $M_1 \supseteq \mathcal{S}_1, M_2 \supseteq \mathcal{S}_2$?

SIM was shown to be \mathcal{NP} -complete in [25]. We use a reduction from SIM as follows. From $G^* = (V, E)$ with subsets \mathcal{S}_1 and \mathcal{S}_2 , we construct a simple graph G by replacing each edge $e = [x, y]$ in \mathcal{S}_i by the graph given in Figure 4.5. We set $P_e = \{[x_e'', y_e'], [y_e', x_e''']\}$ with $h_e^1 = 0, h_e^2 = h_e^3 = 1$ for $i = 1$ or with $h_e^1 = 1, h_e^2 = 0, h_e^3 = 1$ for $i = 2$. Note that in any solution of $\Psi^*(G, 3, \mathcal{P}, H)$ the edge $[x_e', y]$ will get the same color as $[x, y_e']$.

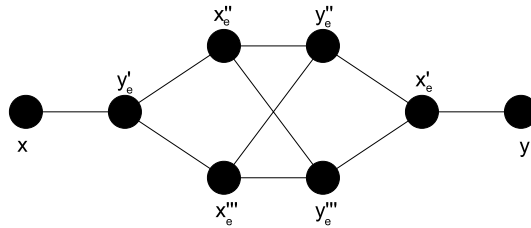


Figure 4.5: Transformation of G where edge $e = [x, y]$ is precolored into G^* .

G is a 3-regular bipartite simple graph; it has an edge 3-coloring satisfying the requirements on each P_e if and only if G^* has an edge 3-coloring where each edge e in \mathcal{S}_i has color i for $i = 1, 2$. ■

Theorem 4.35. $\Psi^*(G, 3, \mathcal{P}, H)$ is \mathcal{NP} -complete when G is a planar bipartite graph with maximum degree $\Delta(G) \leq 3$ and \mathcal{P} is a family of chains P_i of length 2.

Proof: We shall use a transformation from the precoloring extension problem on edges which is shown to be \mathcal{NP} -complete even for planar, 3-regular bipartite graphs [58].

Let $G' = (X \cup Y, E)$ be a planar 3-regular bipartite graph in which some edges are precolored using colors 1, 2 and 3. For each vertex $i \in X \cup Y$ incident to two precolored edges, color the third edge with the remaining color (if there is a contradiction, the problem has no solution). For each vertex $i \in X \cup Y$ incident to one precolored edge $[i, f]$, take $P_i = \{[i, l], [i, p]\}$ where l and p are the endpoints of the two uncolored edges incident to i . If $[i, f]$ has color $j \in \{1, 2, 3\}$, take $h_i^j = 0, h_i^q = 1, q \neq j, q \in \{1, 2, 3\}$. Delete the precolored edges. We get a planar bipartite graph G with maximum degree $\Delta(G) \leq 3$ and \mathcal{P} is a family of chains P_i of length 2.

It is clear that the precoloring extension problem on the edges of G' has a positive answer if and only if $\Psi^*(G, 3, \mathcal{P}, H)$ has a positive answer. As G can be obtained from G' in polynomial time, we proved that our problem is \mathcal{NP} -complete. ■

4.5 Summary and conclusion

We have studied an extension of the basic image reconstruction problem of discrete tomography. The complexity status of some variations has been determined; the results are summarized in Table 4.1 for $\Lambda(G, k, \mathcal{P}, H)$. Then Table 4.2 presents the results for the case of proper colorings ($\Lambda^*(G, k, \mathcal{P}, H)$).

G	k	$ P_i $	h_i^j	$ P_i \cap P_f $		Status	Theorem
	2				$c(\mathcal{P}) = 2$	P	4.7
	2				$\text{Nest}(\mathcal{P}) = 2$	P	4.8
	2				$ P_i \cap \bigcup P_f \leq 2$ $i \neq f$	P	4.9
dir. tree	2				P_i : directed path	P	4.10
		≤ 2				P	4.11
tree				≤ 1		P	4.15
cactus				≤ 1		P	Prop. 4.17
	2	3			\mathcal{P} 3-regular	NPC	4.2
	2				$\text{Nest}(\mathcal{P}) = 3$ $= c(\mathcal{P})$	NPC	4.3
bipartite	2		≤ 3	≤ 1	$\Delta(G) \leq 4$	NPC	4.4
tree	2				$\Delta(G) = 3$	NPC	4.5
tree	2	≤ 4	≤ 3	≤ 2	diameter ≤ 4	NPC	4.6
	3	3	1		$c(\mathcal{P}) = 2$	NPC	4.12

 Table 4.1: Summary of the results for $\Lambda(G, k, \mathcal{P}, H)$.

G	k	$ P_i $	h_i^j	$ P_i \cap P_f $		Status	Theorem
	2					P	Fact 4.18
		2		≤ 1		P	Rem. 4.27
tree			≤ 1	≤ 1		P	Cor. 4.19
chain				≤ 1	CS property	P	4.25
tree	3				$\Delta(G) = 3$	NPC	4.28
bipartite planar	3	≤ 3	≤ 1	≤ 1		NPC	4.29

 Table 4.2: Summary of the results for $\Lambda^*(G, k, \mathcal{P}, H)$.

Finally for edge k -colorings, Table 4.3 (resp. Table 4.4) shows the status of some problems for arbitrary edge k -colorings, i.e. for $\Psi(G, k, \mathcal{P}, H)$ (resp. for proper edge k -colorings, i.e. for $\Psi^*(G, k, \mathcal{P}, H)$).

G	k	$ P_i $	h_i^j	$ P_i \cap P_f $		Status	Theorem
		≤ 2				P	4.30
tree	2				P_i : chain or bundle; $\Delta(G) = 3$	NPC	4.31
cactus	2				$\Delta(G) = 3$; G triangulated	NPC	4.32

Table 4.3: Summary of the results for $\Psi(G, k, \mathcal{P}, H)$.

G	k	$ P_i $	h_i^j	$ P_i \cap P_f $		Status	Theorem
	3	3	1	0	G 3-regular; P_i : triangle	NPC	4.33
bipartite	3	2		0	G 3-regular	NPC	4.34
bipartite planar	3	≤ 2			$\Delta(G) \leq 3$	NPC	4.35

Table 4.4: Summary of the results for $\Psi^*(G, k, \mathcal{P}, H)$.

There are more cases to examine and it would in particular be interesting to consider a family \mathcal{P} of chains with less restrictive hypotheses in some special classes of graphs. But the results obtained here seem to show that the problems become difficult even in very simple cases.

Chapter 5

Graph coloring with cardinality constraints on the neighborhoods

Introduction

Various extensions of the basic graph coloring model (see [9]) have been studied by many authors from a theoretical point of view and also with a motivation stemming from applications in communication systems, operations scheduling, course timetabling, tomography, etc.

Here we shall consider a few variations of the vertex coloring problem which consist essentially in restricting the number of occurrences of the different colors in a given collection \mathcal{P} of subsets P_i of vertices.

In Chapter 4, a formulation extending the basic image reconstruction problem in discrete tomography was discussed where the subsets P_i were chains in the underlying graph G . It was motivated by a simple maintenance scheduling problem in a city metro network.

In this Chapter we will essentially be interested in colorings, i.e., partitions of the vertex set of a graph, such that in some generalized neighborhood of each vertex x , the number of occurrences of each color i is a given integer h_x^i .

More precisely we are given an undirected connected graph $G = (V, E)$ with n vertices and m edges. Given two vertices x and y , we denote by $d(x, y)$ the distance between x and y (the length of a shortest $x - y$ chain). We denote by $N_d(x)$ the d -neighborhood of $x \in V$, that is the set of vertices y such that $d(x, y) = d$. In case where $d = 1$ we simply write $N(x)$ for the 1-neighborhood (or neighborhood, as usual) of x , i.e., the set of vertices y such that $[x, y] \in E$. We also define $N_d^+(x) = \bigcup_{0 \leq l \leq d} N_l(x)$ as the set of vertices at distance at most d from x (with $N_0(x) = \{x\}$).

We are also given a set $\{1, 2, \dots, k\}$ of colors as well as a set $H = \{h(x) = (h_x^1, \dots, h_x^k) \in \mathbb{N}^k | x \in V\}$.

In the first problem, we have to find a k -partition V^1, V^2, \dots, V^k of V such that

$$|N(x) \cap V^i| = h_x^i \text{ for all } x \in V \text{ and all } 1 \leq i \leq k. \quad (5.1)$$

We call this problem $\mathcal{P}(G, H, k)$. In addition, in case we want to obtain a proper coloring (two adjacent vertices must be in two distinct sets V^i and V^j) we let $\mathcal{P}^*(G, H, k)$ denote the corresponding problem.

We will also study the *bounded* version of these problems: we have to find a k -partition V^1, V^2, \dots, V^k of V such that

$$|N(x) \cap V^i| \leq h_x^i \text{ for all } x \in V \text{ and all } 1 \leq i \leq k. \quad (5.2)$$

We will call these problems $\mathcal{BP}(G, H, k)$ and $\mathcal{BP}^*(G, H, k)$, respectively.

Our second problem is to find a k -partition V^1, V^2, \dots, V^k of V such that

$$|N^+(x) \cap V^i| = h_x^i \text{ for all } x \in V \text{ and all } 1 \leq i \leq k. \quad (5.3)$$

We call this problem and its proper coloring version $\mathcal{P}_1^+(G, H, k)$ and $\mathcal{P}_1^{+*}(G, H, k)$, respectively.

We will also be interested in $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$, the problems of finding a k -partition, respectively a proper coloring, V^1, V^2, \dots, V^k of V such that

$$|N_2(x) \cap V^i| = h_x^i \text{ for all } x \in V \text{ and all } 1 \leq i \leq k. \quad (5.4)$$

Notice that our formulation includes the so called cardinality constrained coloring problem which consists in determining if a graph $G = (V, E)$ has a proper k -coloring (V^1, \dots, V^k) with given cardinality s_i for each color class V^i (see [5, 14, 26, 40, 46] for results on this problem): it suffices to take any d larger than or equal to the diameter of G in the set $N_d^+(x)$ defined above (since then $\bigcup_{l=0}^d N_l(x) = V$ for each x) with $h_x^i = s_i$ for all x and all $1 \leq i \leq k$.

These problems are close to the well known $L(h, k)$ -Labelling problems (see [12] for a survey). The problem consists in an assignment of nonnegative integers (i.e., colors) to the vertices of a graph such that adjacent vertices get colors which differ by at least h and vertices joined by a chain of length two receive colors differing by at least k (even if there is an edge joining these vertices). Applications to channel assignment or to multihop radio networks are mentioned in [12]. Under the assumption $h_x^i = 1$, for all i and for all x , the colorings of $\mathcal{BP}^*(G, H, k)$ and those of $L(1, 1)$ -Labelling satisfy the same requirements: adjacent vertices have different colors and vertices linked by a chain of length two (i.e. common neighbors of a single vertex) have different colors. It is also close to the so called star coloring problem studied in [32].

One should also recall that nonproper coloring models have been used under the name of defective coloring in [22] in a frequency assignment context where interferences had to be minimized. Applications to scheduling are also discussed there.

Let us denote by $s(x) = \{i : h_x^i > 0\}$, $x \in V$, the set of colors required to occur in $N(x)$. We have the following facts which will be used implicitly in the algorithms of the next sections.

Fact 5.1. *If $\mathcal{P}(G, H, k)$ has a solution, then $\bigcap_{y \in N(x)} s(y) \neq \emptyset$ for all $x \in V$.*

Fact 5.2. *If for a given $x \in V$, $\bigcap_{y \in N(x)} s(y) = \{i\}$, then in any solution of $\mathcal{P}(G, H, k)$ we have $x \in V^i$.*

Notice that these facts also hold for $\mathcal{P}_1^+(G, H, k)$.

Fact 5.3. *If $\mathcal{P}_1^{+*}(G, H, k)$ has a solution, then for every vertex x there is a color i such that $h_x^i = 1$.*

Fact 5.4. *If $\mathcal{P}_1^{+*}(G, H, k)$ has a solution, then for each color i and each vertex x such that $h_x^i \neq 1$ we have $x \notin V^i$.*

5.1 \mathcal{NP} -completeness results

We shall study here the complexity status of problems $\mathcal{P}(G, H, 2)$, $\mathcal{P}^*(G, H, 3)$, $\mathcal{BP}^*(G, H, 3)$, $\mathcal{BP}^*(G, H, 4)$, $\mathcal{P}_1^+(G, H, 2)$ and $\mathcal{P}_1^{+*}(G, H, 3)$.

Theorem 5.5. *$\mathcal{P}(G, H, 2)$ is \mathcal{NP} -complete even if G is a 3-regular planar bipartite graph.*

Proof: We use a transformation from the CUBIC PLANAR MONOTONE 1-in-3SAT problem which is known to be \mathcal{NP} -complete (see [60]). In this problem we are given a set X of variables and a set C of clauses of the form $(a \vee b \vee c)$ where a, b and c are distinct variables without negation such that the underlying bipartite graph $G = (X \cup C, E) = (X \cup C, \{[x_i, \hat{c}] | x_i \text{ occurring in clause } \hat{c} \in C\})$ is 3-regular and planar. The question is to decide whether there exists a truth assignment such that exactly one variable in each clause is ‘true’.

Consider an instance of CUBIC PLANAR MONOTONE 1-in-3SAT as well as its corresponding graph G . For each vertex \hat{c} , representing a clause, we set $h(\hat{c}) = (1, 2)$ and for each vertex x , representing a variable x , we set $h(x) = (3, 0)$.

Consider a positive instance of CUBIC PLANAR MONOTONE 1-in-3SAT. Then for each variable x , if x is ‘true’, we assign x to V^1 and if x is ‘false’, we assign x to V^2 . All the vertices representing clauses are assigned to V^1 . Thus we get a positive answer for the corresponding instance of $\mathcal{P}(G, H, 2)$. Conversely, if an instance of $\mathcal{P}(G, H, 2)$ is positive, then by setting x to ‘true’ if x has color 1 and to ‘false’ if x has color 2, the corresponding instance of CUBIC PLANAR MONOTONE 1-in-3SAT is true: all vertices corresponding to clauses \hat{c} are in V^1 since $h(x) = (3, 0)$ for all vertices x . Every x will be in V^1 or V^2 . Since $h(\hat{c}) = (1, 2)$, clause \hat{c} will have exactly one variable x occurring in V^1 , i.e., one variable which is ‘true’. ■

Theorem 5.6. *$\mathcal{P}^*(G, H, 3)$ is \mathcal{NP} -complete even if G is 3-regular planar bipartite.*

Proof: We use the same reduction as in the proof of Theorem 5.5 except that we take $h(x) = (0, 0, 3)$ for each vertex x representing a variable and $h(\hat{c}) = (1, 2, 0)$ for each vertex \hat{c} representing a clause. Given a positive instance of CUBIC PLANAR MONOTONE 1-in-3SAT, each variable x which is ‘true’ is assigned to V^1 ; it is assigned to V^2 if it is ‘false’. All clauses \hat{c} are assigned to V^3 . So we obtain a feasible solution of $\mathcal{P}^*(G, H, 3)$. Conversely if an instance of $\mathcal{P}^*(G, H, 3)$ is positive, all vertices \hat{c} corresponding to clauses are in V^3 since $h(x) = (0, 0, 3)$ for each x representing a variable. Since $h(\hat{c}) = (1, 2, 0)$ exactly one variable x occurring in \hat{c} will be ‘true’ (x will be in V^1) and two variables in \hat{c} will be ‘false’. This will give a positive instance of CUBIC PLANAR MONOTONE 1-in-3SAT. ■

Theorem 5.7. $\mathcal{BP}^*(G, H, 4)$ is \mathcal{NP} -complete even if G is a bipartite graph with maximum degree 3 and $h_x^i = 1 \forall x \in V, i = 1, 2, 3, 4$.

Proof: We use a reduction from the edge 3-coloring problem of a 3-regular graph. This problem is known to be \mathcal{NP} -complete (see [43]).

Let G' be a 3-regular graph. For each vertex x of G' we introduce the vertex gadget including (among others) vertices x_1, x_2, x_3 and x_4 shown in Figure 5.1; each edge $[x, y]$ of G' corresponds to a unique edge $[x_u, y_v]$ in the new graph. We locally replace every edge $[x_u, y_v]$ by the edge gadget $J(x_u, y_v)$ given in Figure 5.2. The resulting graph G is bipartite and has maximum degree 3. A vertex gadget has the following properties: vertices x_1, \dots, x_4 must all have different colors. W.l.o.g. we may assume initially that vertex x_4 has color 4, vertices x_1, x_2 and x_3 will then have different colors in $\{1, 2, 3\}$. Each x_i ($i \leq 3$) with color, say $c(i)$, will have its two neighbors, s and t , in the vertex gadget with colors $c(s) \neq c(t)$ and $c(s), c(t) \in \{1, 2, 3\} \setminus c(i)$. We will show later that this will indeed hold for every vertex gadget in G . For any edge gadget $J(x_u, y_v)$, we have $c(x_u) = c(y_v)$ and $c(a) = c(a') = 4$, where a and a' are shown in Figure 5.2. This holds because if x_u has a color $c(x_u)$ in $\{1, 2, 3\}$, say color 1, then since x_u is in a vertex gadget (its two neighbors in the vertex gadget have colors 2 and 3) a has color 4, this implies colors 2 and 3 for b and c w.l.o.g., and d gets color 1; then e must receive color 4. It also follows that b' and c' have w.l.o.g. colors 3 and 2. Now if d' gets color 4 we reach a contradiction (e' gets color 1 and this is not possible if e has color 4). So d' must have color 1, which gives color 4 for e' and color 1 for y_v . It now follows that we will have color 4 for each vertex x_4 in the vertex gadgets of vertices x and so the vertices x_1, x_2 and x_3 of every vertex gadget associated with x will all have different colors in $\{1, 2, 3\}$.

Suppose that an instance of $\mathcal{BP}^*(G, H, 4)$ has a solution ‘true’. By coloring each edge $[x, y]$ in G' with the color of the corresponding vertices x_u and y_v in G (remember that these two vertices necessarily have the same color $c \in \{1, 2, 3\}$), we get a feasible 3-coloring of the edges of G' .

Now suppose that we have a 3-coloring of the edges of G' . If an edge $[x, y]$ has color $c \in \{1, 2, 3\}$, then color the corresponding vertices x_u and y_v in G with color c . Once we have done this for all the edges in G' , we can complete as explained above the coloring using

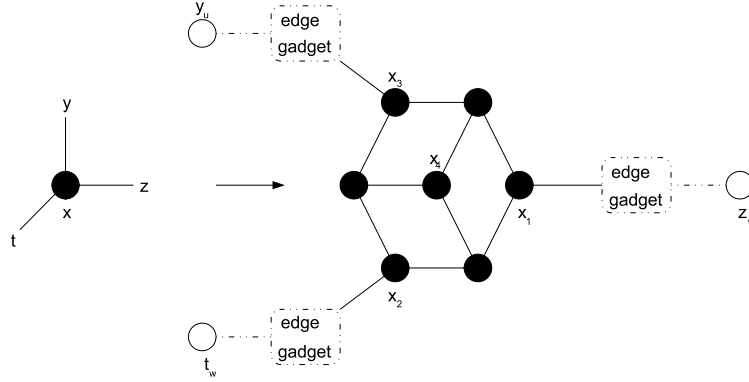


Figure 5.1: The vertex gadget replacing a vertex x .

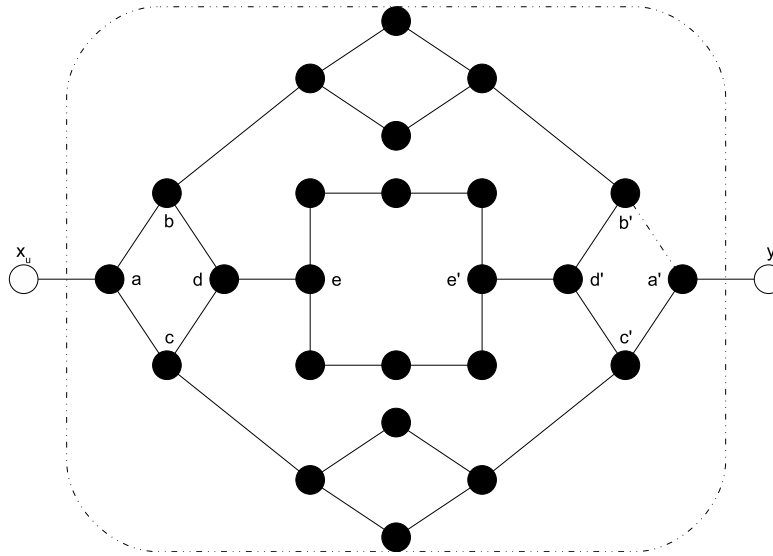


Figure 5.2: The edge gadget $J(x_u, y_v)$ corresponding to an edge $[x_u, y_v]$.

at most 4 colors and satisfying $|N(x) \cap V^i| \leq h_x^i = 1 \forall x \in V, i = 1, 2, 3, 4$. ■

It is shown in [12] that $L(1,1)$ is \mathcal{NP} -hard in bipartite graphs; the associated coloring problem with $k = 4$ colors is shown to be \mathcal{NP} -complete in 3-regular graphs. We have the following consequence of Theorem 5.7.

Corollary 5.8. *$L(1,1)$ is \mathcal{NP} -complete even in bipartite graphs with maximum degree 3 and 4 colors.*

We will need the following Lemma in the proof of Theorem 5.10.

Lemma 5.9. *$\mathcal{BP}^*(G, H, 3)$ is \mathcal{NP} -complete even if G is planar with maximum degree 4 and $h_x^i = 2 \forall x \in V, i = 1, 2, 3$.*

Proof: We use a reduction from the problem of 3-coloring a planar graph with maximum

degree 4. This problem is known to be \mathcal{NP} -complete (see [23]). Let G' be a planar graph with maximum degree 4. We replace each vertex x by the vertex gadget shown in Figure 5.3 and an edge $[x, y]$ in G' will be replaced by a suitable edge $[x_u, y_v]$, $u, v \in \{1, 2, 3, 4\}$. We obtain a planar graph G with maximum degree 4.

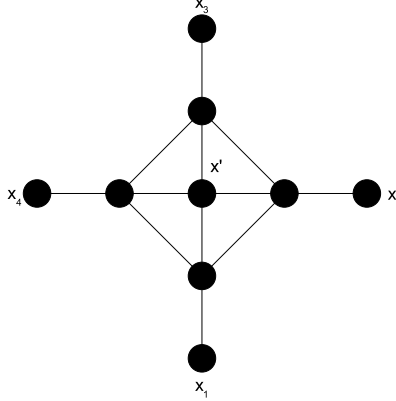


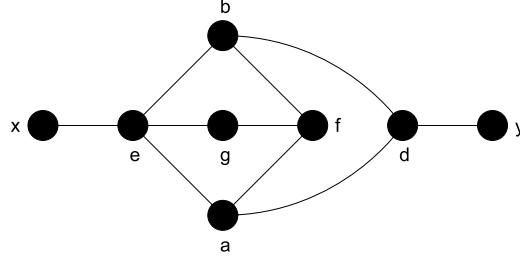
Figure 5.3: The vertex gadget replacing a vertex x .

Now suppose that there is a 3-coloring of G such that $|N(x) \cap V^i| \leq 2 \forall x \in V, i = 1, 2, 3$. Necessarily x_1, x_2, x_3 and x_4 must be colored with the same color as x' . Coloring the corresponding vertex x in G' with this color will give us a 3-coloring of G' .

Conversely, suppose we have a 3-coloring of the vertices of G' . If x has color c , then color the corresponding vertices x', x_1, x_2, x_3 and x_4 with this same color c in G . Then the remaining vertices can be colored using 3 colors in such a way that $|N(x) \cap V^i| \leq 2 \forall x \in V, i = 1, 2, 3$. So we get a positive solution for the instance of $\mathcal{BP}^*(G, H, 3)$. ■

Theorem 5.10. $\mathcal{BP}^*(G, H, 3)$ is \mathcal{NP} -complete even if G is planar bipartite with maximum degree 4 and $h_x^i = 2 \forall x \in V, i = 1, 2, 3$.

Proof: We use a transformation from $\mathcal{BP}^*(G', H, 3)$ which is \mathcal{NP} -complete when G' is planar with maximum degree 4 and $h_x^i = 2 \forall x \in V, i = 1, 2, 3$, as shown in Lemma 5.9. Let G' be a planar graph with maximum degree 4. We replace each edge $[x, y]$ by the edge gadget shown in Figure 5.4. We obtain a planar bipartite graph G with maximum degree 4. Now suppose that there is a 3-coloring of G such that $|N(x) \cap V^i| \leq h_x^i = 2 \forall x \in V, i = 1, 2, 3$. Denote by c this coloring. We must have $c(a) = c(b)$, since otherwise all vertices in $N(a) \cap N(b)$ should have the same color, which would violate the requirements on $h_a^i = h_b^i = 2$; similarly $c(e) = c(f)$. So let $c(a) = c(b) = 1$ and $c(e) = c(f) = 2$. We must have $c(g) = c(x) = 3$; then $c(d) \neq c(a) = 1$ since $d \in N(a)$, and $c(d) \neq c(f) = 2$ since $h_a^2 = 2$, so $c(d) = 3 = c(x) = c(g)$. Finally $c(y) \neq c(d) = 3$ ($y \in N(d)$), $c(y) \neq 1$ (since $h_d^1 = 2$), so $c(y) = 2 = c(e) = c(f)$. Thus x and y get different colors. Coloring the vertices x and y in G' with the color they get in G , we obtain a 3-coloring of G' . In fact, since $c(e) = c(y)$ and $|N(x) \cap V^i| \leq 2, i = 1, 2, 3$, in G , we will obtain a solution in G' satisfying

Figure 5.4: The edge gadget replacing an edge $[x, y]$.

the constraints $|N(x) \cap V^i| \leq 2 \forall x \in V, i = 1, 2, 3$.

Conversely, suppose that there is a 3-coloring of G' with $|N(x) \cap V^i| \leq 2 \forall x \in V, i = 1, 2, 3$. Then by coloring the corresponding vertices in G with the same colors and by applying the rules mentioned above for the remaining vertices, we get a feasible 3-coloring of G . ■

Theorem 5.11. $\mathcal{P}_1^+(G, H, 2)$ is \mathcal{NP} -complete even if G is planar bipartite of maximum degree 4.

Proof: We use a transformation from $\mathcal{P}(G', H, 2)$ for a 3-regular planar bipartite graph G' (see Theorem 5.5). From G' we build a graph G as follows. For each vertex x' of G' , we introduce a new vertex x ; x and x' are linked by the edge $[x, x']$; every edge $[x', y']$ of G' is also an edge of G . Thus G is planar bipartite with maximum degree 4. Now, for each new vertex x we set $h(x) = (1, 1)$, and if we have $h(x') = (a, b)$ in the instance of $\mathcal{P}(G', H, 2)$ we set $h(x') = (a + 1, b + 1)$ for its corresponding instance $\mathcal{P}_1^+(G, H, 2)$. Let V^1, V^2 be a 2-coloring of G' , then we obtain a 2-coloring for G as follows: the twin x of x' is introduced into V^2 if $x' \in V^1$, and vice versa. Conversely, if we have a 2-coloring of G , then by deleting the new vertices we obtain a 2-coloring of G' . ■

Theorem 5.12. $\mathcal{P}_1^{+*}(G, H, 3)$ is \mathcal{NP} -complete even if G is planar bipartite of maximum degree 4.

Proof: We use a reduction from CUBIC PLANAR MONOTONE 1-in-3SAT. Let G be the 3-regular planar bipartite graph associated with this problem. For each vertex x in G representing a variable, we introduce a new vertex x' and an edge $[x, x']$. We obtain a planar bipartite graph with maximum degree 4. We set $h(x) = (1, 1, 3)$, $h(x') = (1, 1, 0)$ and for the vertices \hat{c} representing the clauses we set $h(\hat{c}) = (1, 2, 1)$.

Suppose that an instance of CUBIC PLANAR MONOTONE 1-in-3SAT has a solution 'true'. Then for each variable x which is 'true', we assign x to V^1 and x' to V^2 , and for each variable x which is 'false', we assign x to V^2 and x' to V^1 . All the vertices \hat{c} representing a clause are assigned to V^3 . Thus we get a positive answer to the corresponding instance of $\mathcal{P}_1^{+*}(G, H, 3)$.

Conversely, assume an instance of $\mathcal{P}_1^{+*}(G, H, 3)$ has a value ‘true’; then, since $h(x') = (1, 1, 0)$ vertices x and x' cannot be in V^3 ; one will be in V^1 , the other in V^2 . Since every x must have exactly three neighbors in V^3 , all vertices \hat{c} representing clauses are necessarily in V^3 . Setting x to ‘true’ if x has color 1 and to ‘false’ if x has color 2, we get a positive answer to the instance of CUBIC PLANAR MONOTONE 1-in-3SAT. ■

5.2 The special case of trees

We shall now give a general dynamic programming algorithm which will show that $\mathcal{P}(G, H, k)$, $\mathcal{P}^*(G, H, k)$, $\mathcal{P}_1^+(G, H, k)$, $\mathcal{P}_1^{+*}(G, H, k)$ and their bounded versions can be solved in polynomial time when G is a tree. A version adapted to $\mathcal{P}(G, H, k)$ will be described and we will show later how it can be modified to handle the other problems.

We consider a tree $T = (V, E)$ on n vertices. We root T at an arbitrary vertex r . For any vertex x of T we denote by $T(x)$ the subtree of T rooted at vertex x . By extension $T(x)$ will also be the set of vertices in $T(x)$. Let y be the father of a vertex x in T , $x \neq r$. We denote the set of possible colors for a vertex x by $L(x) \subseteq C = \{1, 2, \dots, k\}$ in the given problem. We set $L(x) = \bigcap_{z \in N(x)} s(z)$.

We introduce a function $F : V \times V \times C \times C \longrightarrow \{0, 1\}$, defined when (y, x) is an arc of the rooted tree and the pair of colors belongs to $L(x) \times L(y)$. Depending on the nature of vertex x , the function F is defined recursively as follows.

- (1) if x is a leaf, $F(x, y, c, c') = 1$ iff $h_x^{c'} = 1$ and $h_y^c > 0$;
- (2) if x is not a leaf, $F(x, y, c, c') = 1$ iff $\forall z \in N(x) \cap T(x)$ there exists a color c'' such that $F(z, x, c'', c) = 1$ and there exists a partition U_1, U_2, \dots, U_k of $N(x) \cap T(x)$ such that
 - (a) $|U_i| = |\{z \in N(x) \cap T(x) | z \text{ is colored with color } i\}| = h_x^i$ if $i \neq c'$;
 - (b) $|U_{c'}| = h_x^{c'} - 1$.

While in the first case, $F(x, y, c, c')$ can be determined in constant time, in the second case we shall use an auxiliary graph $B(x)$ in order to find the required partition. For $x \neq r$ we construct the bipartite graph $B(x) = (V_1, V_2, U)$ with $V_1 = N(x) \cap T(x)$ representing the neighbors of x contained in $T(x)$ and V_2 is defined as follows. For each $i \in L(x)$ set $V_2(i) = \{i_j | j = 1, \dots, h_x^i\}$ and $V_2 = (\bigcup_{i \in L(x), i \neq c'} V_2(i)) \cup \{c'_l | l = 1, \dots, h_x^{c'} - 1\}$. This will represent the occurrences of the different colors (for each color i different from c' we introduce h_x^i vertices and for color c' we introduce $h_x^{c'} - 1$ vertices) in $N(x) \cap T(x)$. We introduce an edge $[z, i_j]$ if $F(z, x, i, c) = 1$. In this graph $B(x)$, a perfect matching will correspond to a

partition U_1, \dots, U_k of $N(x) \cap T(x)$ satisfying the required constraints.

For the root r of T we define the following function $\Pi : \{r\} \times L(r) \rightarrow \{0, 1\}$, $\Pi(r, c) = 1$ iff $\forall z \in N(r)$ there exists a color c'' such that $F(z, r, c'', c) = 1$ and there exists a partition U_1, U_2, \dots, U_k of $N(r)$ such that $|U_i| = h_r^i \forall i$. Again, as described above, a perfect matching in an auxiliary bipartite graph $B(r)$ will give us a partition satisfying the constraints.

Thus we get the following algorithm.

Algorithm

- (1) Number the vertices in reverse order of Breadth First Search (the leaves come first, the root is at the end). Let x_1, \dots, x_n be the vertices.
- (2) For $i = 1$ to n and for each pair of colors $(c, c') \in L(x_i) \times L(y)$, compute $F(x_i, y, c, c')$ where y is the father of x_i . If $F(x_i, y, c, c') = 0$ for each pair (c, c') , the problem has no solution.
- (3) Construct the feasible coloring of $\mathcal{P}(T, H, k)$ starting from the root r and recalling the pairs (c, c') for which $F(x_i, y, c, c') = 1$.

Theorem 5.13. *The above algorithm solves problem $\mathcal{P}(T, H, k)$ in $O(k^2 n^{2.5})$ time.*

Proof: When $F(x, y, c, c') = 1$ it means that there is a feasible solution for the problem associated with the subtree $T(x)$ where x has color c and its father y has color c' . Since for each x , all pairs (c, c') are examined we will obtain a solution whenever there exists one. If $\Pi(r, c) = 1$, assign color c to r ; then for each arc (y, x) where y is colored with color c (x is not colored) and $F(x, y, c', c) = 1$, assign color c' to x ; x is then colored.

Let us now analyze the complexity of this dynamic programming approach. For each vertex x in T we have $O(k^2)$ pairs of colors (c, c') for which we have to determine the value of F . A perfect matching can be determined in $O(n^{2.5})$ in a bipartite graph with $2n$ vertices (see [1]). In our case the auxiliary bipartite graph $B(x)$ which we construct for a vertex x of T contains $2(d(x) - 1)$ vertices, where $d(x) = |N(x)|$, and hence a perfect matching can be computed in $O(d(x)^{2.5})$ time. Thus the values of F for each vertex and each pair of colors can be obtained in $O(k^2 \sum_{x \in T} d(x)^{2.5})$ time, i.e., our algorithm has a complexity of $O(k^2 n^{2.5})$. ■

We will now explain how the previous algorithm can be adapted to the problems $\mathcal{P}^*(G, H, k)$, $\mathcal{P}_1^+(G, H, k)$, $\mathcal{P}_1^{+*}(G, H, k)$ and their bounded versions.

- $\mathcal{P}^*(G, H, k)$

We just have to add the constraint that $c \neq c'$ in the definition of F ; in this way we

avoid having two adjacent vertices which will be colored with the same color.

- $\mathcal{P}_1^+(G, H, k)$

First we have to adapt the definition of $L(x)$, i.e., $L(x) = \bigcap_{z \in N^+(x)} s(z)$. Then we must modify the computation of F in the following way.

(1) if x is a leaf, $F(x, y, c, c') = 1$ iff

(a) $h_x^c = h_x^{c'} = 1$, with $c \neq c'$, and $h_y^c, h_y^{c'} \geq 0$

or

(b) $h_x^c = 2$, with $c = c'$ and $h_y^c \geq 0$;

(2) if x is not a leaf, $F(x, y, c, c') = 1$ iff

$\forall z \in N(x) \cap T(x)$ there exists a color c'' such that $F(z, x, c'', c) = 1$ and there exists a partition U_1, U_2, \dots, U_k of $N(x) \cap T(x)$ such that

(a) $|U_i| = h_x^i$ if $i \neq c, c'$;

(b) $|U_c| = h_x^c - 1$, and $|U_{c'}| = h_x^{c'} - 1$, if $c \neq c'$;

(c) $|U_c| = h_x^c - 2$, if $c = c'$.

Finally we also have to modify the definition of Π : $\Pi(r, c) = 1$ iff $\forall z \in N(r)$ there exists a color c'' such that $F(z, r, c'', c) = 1$ and there exists a partition U_1, U_2, \dots, U_k of $N(r)$ such that

(1) $|U_i| = h_r^i$ if $i \neq c$;

(2) $|U_c| = h_r^c - 1$.

In the auxiliary graph $B(x)$ constructed as before we introduce $h_x^c - 1$ vertices for color c (instead of h_x^c as used in $\mathcal{P}(G, H, k)$).

- $\mathcal{P}_1^{+*}(G, H, k)$

We use the version for $\mathcal{P}_1^+(G, H, k)$ and add the constraint that $c \neq c'$ in the definition of F .

- For all bounded problems \mathcal{BP} , we adapt the above procedure as follows. Instead of constructing a perfect matching in $B(x)$, we simply determine a matching saturating all vertices in V_1 . It need not be a perfect matching since we must have at most h_x^i vertices of color i in the neighborhood of x but not necessarily exactly h_x^i .

5.3 The case of $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$

Here we will consider a special case of trees for which $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$ can be solved in linear time. We will first give conditions of a solution for a star. We recall that a star $S(y; x_1, \dots, x_n)$ is a tree with $n \geq 2$ such that $E = \{[y, x_i] : 1 \leq i \leq n\}$. y is the center of the star and the x_i 's are the external vertices.

Proposition 5.14. *Given a star $S(y; x_1, \dots, x_n)$ with a collection H of nonnegative integral vectors $h(x) = (h_x^1, h_x^2, h_x^3, \dots, h_x^k)$ for each external vertex x , the following statements are equivalent:*

- (a) $\{x_1, \dots, x_n\}$ has a unique coloring with h_i vertices of color i ;
- (b) (1) for each external vertex x , $h_x^1 + h_x^2 + h_x^3 + \dots + h_x^k = n - 1$;
 (2) for each color i ,
 $n - h_i$ external vertices x have $h_x^i = h_i$ and
 h_i vertices x have $h_x^i = h_i - 1$;
- (3) for each color i let $V(i) = \{x | h_x^i = h_i - 1\}$; then $V(i) \cap V(j) = \emptyset$ for all i, j with $i \neq j$.

Proof: (a) \Rightarrow (b): $\sum_{i=1}^k h_x^i$ is the number of colors (with their multiplicities) which have to occur at distance two from x . Since $|N_2(x)| = n - 1$ for each external vertex x , so (1) holds. An external vertex of color i (resp. color $j \neq i$) will have $h_i - 1$ (resp. h_i) vertices at distance two with color i , so (2) will hold. The set of external vertices with color i will be $V(i)$ and (3) holds.

(b) \Rightarrow (a): For each i we color the h_i vertices x of $V(i)$ with color i and this will give us the required coloring which is uniquely defined. \blacksquare

Remark 5.15. *If G is a star, then the treatments of $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$ are similar. We just have to assign any color $c \in \{1, \dots, k\}$ to the central vertex y for $\mathcal{P}_2(G, H, k)$ and any color $c \in \{1, \dots, k\}$ not used in $N(y)$ (if there is one) for $\mathcal{P}_2^*(G, H, k)$.*

Remark 5.16. $\mathcal{P}_2(G, H, k)$ when G is a star with $n \geq 2$ external vertices, is the same problem as $\mathcal{P}(G', H, k)$ when G' is a complete graph of order n ; if we consider the pairs of external vertices x_p, x_q ($1 \leq p, q \leq n$) in a star, they are all at distance two. In a complete graph G' all pairs of vertices are at distance one. Hence the announced equivalence.

For a special case of trees, we give a complete description of a simple algorithm which will determine in linear time whether a solution exists or not for $\mathcal{P}_2(G, H, k)$.

We define a *quaternary tree* (or shortly *quatree*) as a tree where all internal vertices (i.e., non leaves) have degree at least four. Let (B, W) be the bipartition of the vertex set V (B is the set of black vertices and W of white vertices).

A *pendent star* $S_h(y; x_0, x_1, \dots, x_n)$ in a quatree Q is the subgraph induced by the vertex set $\{y\} \cup N(y)$ where $N(y) = \{x_0, x_1, \dots, x_n\}$ and x_1, \dots, x_n are leaves of Q . Q being a quatree we have $n \geq 3$. S_h is a star for which at least three external vertices are leaves of Q . Notice that x_0 is generally not a leaf (except when Q itself is a star).

Proposition 5.17. *Let $S_h(y; x_0, x_1, \dots, x_n)$ be a pendent star. A necessary condition for a coloring of $N(y)$ to exist is that for any two external vertices x_p and x_q either $h(x_p) = h(x_q)$*

or $|h_{x_p}^c - h_{x_q}^c| \leq 1$ for each color c and there are exactly two colors, say c and c' , such that $h_{x_p}^c \neq h_{x_q}^c$ and $h_{x_p}^{c'} \neq h_{x_q}^{c'}$.

Proof: As for the case of a star (see proof of Proposition 5.14) in any coloring there is no pair of external vertices x_p, x_q with $|h_{x_p}^c - h_{x_q}^c| \geq 2$ for some color c .

We have necessarily $\sum_{i=1}^k h_x^i = n$, so we cannot have exactly one color c such that $h_{x_p}^c \neq h_{x_q}^c$. Now suppose that there are at least three colors c_1, c_2 and c_3 with $h_{x_p}^{c_i} \neq h_{x_q}^{c_i}$, $i \in \{1, 2, 3\}$. As for the case of a star (see proof of Proposition 5.14), if $h_{x_p}^{c_i} = h_{x_q}^{c_i} - 1$, x_p must have color c_i . It follows that x_p or x_q has at least two distinct colors, which is a contradiction. ■

Proposition 5.18. *Let $S_h(y; x_0, x_1, \dots, x_n)$ be a pendent star. If there is a coloring of S_h , it is unique.*

Proof: Suppose that the condition of Proposition 5.17 is satisfied.

In case $h(x_p) = h(x_q)$ for each $1 \leq p, q \leq n$, each external vertex x has the same color c . Then for each x , $h_x^c = n - 1$ or $h_x^c = n$. In the first case, there is a color $c' \neq c$ such that for each x , $h_x^{c'} = 1$ and thus x_0 must get color c' . In the second case, all external vertices x_0, x_1, \dots, x_n necessarily have color c .

In case there exists two vertices x_p and x_q with $h(x_p) \neq h(x_q)$, there is a color c such that $h_{x_p}^c = h_{x_q}^c - 1$. Thus x_p has necessarily color c . So there is another color c' with $h_{x_p}^{c'} = h_{x_q}^{c'} + 1$ and x_q must have color c' . For each external vertex x_f , $f \neq p, q$, since $h(x_p) \neq h(x_q)$ we have $h(x_f) \neq h(x_p)$ or $h(x_f) \neq h(x_q)$. So as above we obtain the color of vertex x_f . In this way we can assign a color to each external vertex x . If an external vertex x receives two distinct colors, clearly there is no solution. Now from each vector $h(x)$, we determine a unique color of x_0 . If there are distinct colors assigned to x_0 , there is no solution, otherwise we obtain a coloring for x_0, x_1, \dots, x_n and this coloring is unique. ■

Theorem 5.19. $\mathcal{P}_2(Q, H, k)$ can be solved in time $O(kn^2)$ when Q is a quatree. Moreover if there is a coloring, it is unique.

Proof: In the following algorithm, we will start by coloring the vertices of W and a similar second run will color the vertices of B . W.l.o.g. we may remove all black leaves for the first run of the algorithm.

Algorithm

- (1) $G \leftarrow Q$;
- (2) while $G \neq \emptyset$ or G is not a star
 for each pendent star $S_h(y; x_0, x_1, \dots, x_n)$ do
 - (2.1) if condition of Proposition 5.17 is not satisfied then there is no solution;

- (2.2) color x_0, x_1, \dots, x_n according to $h(x_1), \dots, h(x_n)$;
- (2.3) if the coloring fails, there is no solution;
- (2.4) update $h(x_0)$ according to the (unique) coloring constructed;
 $G \leftarrow G \setminus \{y, x_1, \dots, x_n\}$;
- (3) if G is a star, then color x_0, x_1, \dots, x_n ;
 if the coloring fails, then there is no solution.

In step (2.2) the unique coloring is obtained as described in the proof of Proposition 5.18. Applying the algorithm to B , we finally obtain a unique coloring of Q if such a coloring exists.

For each pendent star $S_h(y; x_0, x_1, \dots, x_d)$, the condition of Proposition 5.17 can be checked in time $O(k(d(y))^2)$ and its coloring (Proposition 5.18) can be obtained in time $O(kd(y))$. It follows that the whole complexity is $O(\sum_y k(d(y))^2) = O(kn^2)$ since Q is a quater. ■

From the previous result we conclude the following.

Corollary 5.20. *$\mathcal{P}_2^*(Q, H, k)$ can be solved in time $O(kn^2)$ when Q is a quater. Moreover if there is a coloring, it is unique.*

A (unique) coloring exists if there exist a coloring of the white vertices and a coloring of the black vertices and if both colorings are compatible (no two adjacent vertices get the same color).

We have restricted ourselves to the case of quateres; this has allowed us to obtain a simple linear algorithm. Notice first that if all internal black vertices in a tree have degree two, then the problem of coloring the white vertices is equivalent to $\mathcal{P}_1(G', H', k)$ where G' is the tree obtained by removing each black vertex linked to two white vertices w_1 and w_2 , and introducing an edge $[w_1, w_2]$.

In addition (i.e., besides having all internal black vertices with degree two), if we have a degree at least four for each internal white vertex, then one can solve the coloring problem by using the algorithm of $\mathcal{P}_1(G, H, k)$ for the white vertices and the first run of the algorithm of $\mathcal{P}_2(G, H, k)$ in quateres for the black vertices.

For the general case where G is a tree, the algorithms proposed here do not seem easy to be adapted to handle this case even if a single color class (B or W) has at the same time internal vertices of degree two and internal vertices with degree at least four.

5.4 Conclusion

We have studied some coloring problems related to the basic image reconstruction problem which could be solved in polynomial time for trees or sometimes for a subclass of trees: the quateres. These are generally \mathcal{NP} -complete for more general graphs. Furthermore

we obtained a new complexity result concerning the $L(1,1)$ -labelling problem. It would be interesting to examine the problems $\mathcal{P}_2(G, H, k)$ and $\mathcal{P}_2^*(G, H, k)$ in the case of general trees. Also the bounded version of these problems should be considered even in quatrees.

Chapter 6

Degree-constrained edge partitioning in graphs arising from discrete tomography

This chapter is dedicated to H.J. Ryser in acknowledgement for his seminal work which stated the now famous Ryser conditions exactly 50 years ago.

Introduction

In this chapter our aim is to explore some problems that arise from a graph theoretical formulation of the basic image reconstruction problem in discrete tomography (see Chapter 1). We recall that this problem is defined as follows.

Assume we are given an $(m \times n)$ array $A = (a_{ij})$ where each entry may contain a pixel having one of the colors $1, \dots, k$. With an image we may associate the number h_i^s (resp. v_j^s) of pixels with color s in row i (resp. column j).

The basic image reconstruction problem consists in assigning a color in $\{1, \dots, k\}$ to each entry of A so that in each row i (resp. column j) there are exactly h_i^s (resp. v_j^s) pixels with color s (for all $i \leq m, j \leq n, s \leq k$).

Clearly the values h_i^s and v_j^s must satisfy some (necessary) conditions which have already been introduced in Chapter 1:

$$\sum_{s=1}^k h_i^s = n \quad (i = 1, \dots, m) \quad (6.1)$$

$$\sum_{s=1}^k v_j^s = m \quad (j = 1, \dots, n) \quad (6.2)$$

$$\sum_{i=1}^m h_i^s = \sum_{j=1}^n v_j^s \quad (s = 1, \dots, k) \quad (6.3)$$

It is known that for $k = 2$, one can find if there is or not an image corresponding to values h_i^s and v_j^s satisfying 6.1 - 6.3. Indeed in [65], Ryser gives, for the case $k = 2$, necessary and sufficient conditions to be verified by the values h_i^s and v_j^s for a solution to exist (see Chapter 1). Furthermore these conditions can be checked in polynomial time.

We recall that for $k = 4$, the image reconstruction problem was shown to be \mathcal{NP} -complete [16], while for $k = 3$ its status is to our knowledge still open.

In this chapter we shall essentially consider the case where we have $k = 3$ colors unless stated otherwise.

The graph theoretical model we will associate with this problem is the following (see also Chapter 1 and [18]). Each row i of A corresponds to a vertex i . Let X be the set of these vertices. Similarly each column j of A corresponds to a vertex j and we call Y the set of these vertices. In addition each entry a_{ij} of A corresponds to an edge $[i, j]$ between vertex i in X and vertex j in Y . So we have a complete bipartite graph $K_{X,Y}$ with $|X| = m$, $|Y| = n$. Given the values h_i^s and v_j^s , the reconstruction problem consists in partitioning the edge set $E(G)$ of $G = K_{X,Y}$ into k subsets E^1, \dots, E^k (E^s is the subset of edges which will be given color s) such that for each s ($1 \leq s \leq k$)

$$h_i^s \text{ is the number of edges of } E^s \text{ adjacent to vertex } i \text{ in } X \quad (6.4)$$

$$v_j^s \text{ is the number of edges of } E^s \text{ adjacent to vertex } j \text{ in } Y \quad (6.5)$$

For the rest of the chapter, when working in complete bipartite graphs, we assume that conditions 6.1 - 6.3 are verified as well as the Ryser conditions for each color $s = 1, \dots, k$.

In general no requirement is imposed on the structure of the graphs generated by E^s or by $E^{st} \equiv E^s \cup E^t$ besides satisfying 6.4 - 6.5.

Here we shall first examine some variations where the union of some subsets E^s has to satisfy some additional constraints. We will focus on these subsets and we will not care about the other subsets corresponding to the remaining colors.

Let us observe that from constraints 6.1 - 6.2 we see that there are indeed $k - 1$ independent colors, the last one, say color k , will be the ground color (the number of its occurrences in each row and in each column is entirely determined by the occurrences of the first colors $1, \dots, k - 1$). Since we assume $k = 3$, we will have to determine disjoint sets E^1, E^2 and $E^3 = E - (E^1 \cup E^2)$ will be automatically determined and it will satisfy 6.4 and 6.5.

We shall examine in the next sections the situation where the color classes E^1 and E^2 form together a tree or a collection of disjoint chains.

For all these problems we will then examine the corresponding problem in the case where instead of having an underlying graph G which is bipartite (as was $K_{X,Y}$) we have now a complete graph $G = K_X$ on $|X| = m$ vertices. So we are given for each vertex i in G and each color $s = 1, \dots, k$ a non negative integer h_i^s . Our problem then consists in finding a partition of the edge set $E(G)$ of $G = K_X$ into k subsets E^1, \dots, E^k such that for each color s ($s = 1, \dots, k$):

$$h_i^s \text{ is the number of edges of } E^s \text{ adjacent to vertex } i \text{ in } X \quad (6.6)$$

For a solution to exist the following conditions must hold:

$$\sum_{s=1}^k h_i^s = m - 1 \quad (i = 1, \dots, m) \quad (6.7)$$

$$\sum_{i \in X} h_i^s \text{ is even} \quad (s = 1, \dots, k) \quad (6.8)$$

For the rest of the chapter, when working in complete graphs, we assume that conditions 6.7 and 6.8 are satisfied.

We also discuss the case where G is an oriented graph in Section 6.4. In Section 6.5, we give sufficient conditions based on the maximum degree in E^{12} for a solution to exist for the case of non oriented complete bipartite graphs and simply complete graphs. This exhibits a new solvable case of the basic image reconstruction problem with $k = 3$ colors.

Finally we will consider the problem corresponding to special values (0, 1 or 2) of h_i^s (and v_j^s), i.e., the search of two edge-disjoint chains or cycles going through specified vertices in complete bipartite graphs or simply complete graphs.

6.1 The case where E^{12} is a tree in $K_{X,Y}$

The first problem which we consider can be formulated as follows. We assume that $k = 3$; given a complete bipartite graph $G = K_{X,Y}$ with values $h_i^1, h_i^2, v_j^1, v_j^2$ (for $i = 1, \dots, m$; $j = 1, \dots, n$) find two disjoint subsets E^1 and E^2 of edges of $E(G)$ such that 6.4 - 6.5 hold for $s = 1, 2$, and in addition E^{12} is a tree.

W.l.o.g. we may assume that every vertex in G will be adjacent to some edge of E^{12} (otherwise we simply delete the vertices not adjacent to any edge of E^{12} and consider the remaining graph). This assumption can be stated as follows:

$$V(E^1) \cup V(E^2) = V(G) \quad (6.9)$$

Observe also that $h_i^1 + h_i^2$ (resp. $v_j^1 + v_j^2$) will be the degree of vertex i in X (resp. vertex j in Y) in the tree E^{12} .

To avoid dealing with trivial cases, we shall assume that our problem is not degenerate so that each one of colors 1 and 2 occurs on at least one edge.

We shall first state two lemmas that will be repeatedly used to construct the required subset E^{12} of edges by reducing the number of connected components.

Recoloring Lemma *Let C_1 and C_2 be two connected components of E^{12} satisfying 6.4 - 6.5 and such that C_2 contains at least one cycle.*

Assume one can find an edge $[x_1, y_1]$ in C_1 and an edge $[x_2, y_2]$ belonging to some cycle C of C_2 such that $[x_1, y_1]$ and $[x_2, y_2]$ have the same color (both are in E^1 or both are in E^2).

Then by replacing $[x_1, y_1]$ and $[x_2, y_2]$ by $[x_1, y_2]$ and $[x_2, y_1]$, and by giving them the same color as the removed edges we get a single connected component C' which still satisfies 6.4 - 6.5.

Proof: Since $[x_1, y_2]$ and $[x_2, y_1]$ get the same color as $[x_1, y_1]$ and $[x_2, y_2]$, 6.4 - 6.5 are still satisfied. One verifies that x_2 and y_2 are still connected in $C - [x_2, y_2]$; furthermore in C' , x_1 and y_1 are connected by a chain consisting of edge $[x_1, y_2]$ followed by $C - [x_2, y_2]$ and by edge $[x_2, y_1]$. It follows that there are chains between any two vertices of C' . Notice furthermore that C' is still bipartite if C_1 and C_2 were bipartite. ■

Recycling Lemma *Assume we have a connected component C of E^{12} satisfying 6.4 - 6.5 and containing some cycle \overline{C} ; let e be an edge of C not contained in any cycle. If there is a chain \tilde{C} in C containing e and starting with some edge $[x_2, y_2]$ in \overline{C} and ending with an edge $[x_1, y_1] \neq e$ in $C - \overline{C}$ with the same color as $[x_2, y_2]$, then one may exchange edges so that 6.4 - 6.5 still hold and e is on a cycle.*

Proof: Notice that $[x_1, y_2]$ and $[x_2, y_1]$ are not simultaneously in E^{12} (otherwise e would be in a cycle). Replacing $[x_1, y_1]$ and $[x_2, y_2]$ by $[x_1, y_2]$ and $[x_2, y_1]$, and assigning them the same color as $[x_1, y_1]$, $[x_2, y_2]$ gives another connected component where 6.4 - 6.5 still hold. It can be checked that there is a cycle C' (possibly of length 2) containing e which goes either through x_1 and y_2 or through x_2 and y_1 . ■

Remark 6.1. *We shall use later an oriented version of these lemmas; the translation to the new case will be immediate.*

Proposition 6.2. *In $G = K_{X,Y}$ there exist two disjoint subsets E^1 and E^2 of edges such that 6.4 - 6.5 hold, and E^{12} is a tree if and only if*

$$(a) \sum_{i \in X} (h_i^1 + h_i^2) = \sum_{j \in Y} (v_j^1 + v_j^2) = (m + n - 1);$$

$$(b) \sum_{i \in X} h_i^s = \sum_{j \in Y} v_j^s \leq (m_s + n_s - 1) \text{ for } s = 1, 2 \text{ where } m_s \text{ (resp. } n_s) \text{ is the number of vertices } i \text{ in } X \text{ (resp. } j \text{ in } Y) \text{ with } h_i^s > 0 \text{ (resp. } v_j^s > 0) \text{ for } s = 1, 2.$$

Proof:

\Rightarrow If E^{12} is a tree it does satisfy (a) and E^1, E^2 cannot contain any cycle, so they are forests and (b) is verified.

\Leftarrow From [65] we know how to construct E^1 and E^2 , since the Ryser conditions are satisfied. Notice that some edges may appear in both E^1 and E^2 , creating cycles of length 2. But these will be removed later in the process.

If E^{12} is connected, it is a tree from (a) and 6.9 and we are done.

Otherwise, E^{12} consists of $p \geq 2$ connected components C_1, \dots, C_p . From (a) there is at least one such component, say C_1 , which is a tree and at least one that contains cycles. By 6.9, C_1 contains at least one edge.

As long as we can find two edges $[x_1, y_1]$ and $[x_2, y_2]$ of the same color (1 or 2) in two connected components and such that in addition $[x_1, y_1]$ is in some cycle, we can reduce these components to a single component by the Recoloring Lemma.

When we cannot find such pairs of edges anymore, either we are done or we are necessarily in the following situation. All connected components that are trees are monochromatic and all have the same color, say 1. Furthermore there is exactly one additional connected component C that contains cycles (otherwise we could have used the Recoloring Lemma); all edges belonging to cycles in C have color 2.

Notice that in C there must be at least one edge of color 1, otherwise (b) would be violated for color 2. From (b) we also know that C must also contain an edge $[x_2, y_2]$ of color 2 which is not incident to any cycle of C . It is linked to some vertex x^* of a cycle \overline{C} by a chain Q containing at least one edge e of color 1. Now take some edge $[x_1, y_1]$ of \overline{C} . Applying the Recycling Lemma, we replace $[x_1, y_1]$ and $[x_2, y_2]$ by $[x_1, y_2]$ and $[x_2, y_1]$; it gives a connected component where edge e (of color 1) now belongs to some cycle (which may possibly be of length 2).

Now we are again in the situation where e has color 1 and it belongs to some cycle of a connected component; besides this there is at least one component F which is a tree and where all edges have color 1. So we can apply the Recoloring Lemma to e and some edge e' of F ; this will not create any new cycle of length 2 since the new edges join distinct vertices of two different connected components. With this we reduce the number of connected components. We repeat this until we get a connected E^{12} ; it will be a tree satisfying all requirements. ■

6.2 The case where E^{12} is a collection of vertex disjoint chains in $K_{X,Y}$

We are now given values h_i^s, v_j^s which satisfy

$$1 \leq h_i^1 + h_i^2 \leq 2 \quad \text{for each } i \text{ in } X \quad (6.10)$$

$$1 \leq v_j^1 + v_j^2 \leq 2 \quad \text{for each } j \text{ in } Y \quad (6.11)$$

Here E^{12} will have to consist of a collection of elementary open chains having their endvertices at vertices r (resp. t) with $h_r^1 + h_r^2 = 1$ (resp. $v_t^1 + v_t^2 = 1$). These will be called *odd vertices*. Clearly we must have an even positive number of odd vertices for the existence of a solution.

Notice that we exclude cycles in a solution, i.e., we have to show that we only have open chains.

Proposition 6.3. *In a complete bipartite graph $K_{X,Y}$ there exist subsets E^1 and E^2 of edges satisfying 6.4 - 6.5 and such that E^{12} is a collection of elementary open chains if and only if*

(a) *for each color $s \in \{1,2\}$, there is at least one vertex which has to be adjacent to exactly one edge of color s ;*

(b) *there exists a vertex $i \in X$ with $h_i^1 + h_i^2 = 1$ or a vertex $j \in Y$ with $v_j^1 + v_j^2 = 1$.*

Proof: It follows from (b) and from 6.3 that the number of odd vertices is even and positive.

If (a) does not hold, there is one color s such that every vertex is adjacent to two edges or to no edge of color s . Clearly the edges of color s cannot be on a chain of E^{12} .

To show that the conditions are sufficient, we start from a set E^{12} satisfying 6.10 - 6.11; E^1 and E^2 can be constructed separately since the Ryser conditions are assumed to hold. As before E^{12} may contain cycles of length two. If it contains no connected component which is a cycle we are done. Otherwise consider a cycle C ; since there is at least one odd vertex (from (b)) there is a chain C' in E^{12} ; if C is not monochromatic we can find a pair of edges $[x_1, y_1]$ in C' , $[x_2, y_2]$ in C of the same color and we use the Recoloring Lemma. When we cannot use the Lemma anymore we are in the situation where we have monochromatic cycles (all of the same color, say 1) and monochromatic chains (all of color 2) between odd vertices. But this is not possible: from (a) for color 1, there must be a vertex adjacent to exactly one edge of color 1. Hence we do not have this case and we can construct a set E^{12} satisfying 6.10 - 6.11 and consisting of open chains. ■

6.3 The case where G is a complete graph K_X

Similar problems as in the previous section can be raised for the case where G is simply a complete graph K_X on a set X of m vertices.

Although this has no immediate connection with discrete tomography as before, we mention it for its interest in a graph theoretical context.

As already mentioned in the introduction we now want to find a partition E^1, \dots, E^k of the edge set $E(G)$ of $G = K_X$ such that in each E^s there are exactly h_i^s edges adjacent to vertex i ($i = 1, \dots, m$) (condition 6.6). E^s is usually called a *b-factor*. Since we are in K_X , the conditions of existence are given by the Erdős-Gallai theorem (existence of a simple graph with given degrees; see Chapter 6 in [9]). We shall assume that these conditions hold for each E^s (otherwise our problem has no solution).

Furthermore we assume, as before, that 6.9 holds, i.e., every vertex is adjacent to at least one edge of $E^1 \cup \dots \cup E^{k-1}$.

The following proposition shed some light on the relative complexity of the decision versions of the edge k -partitioning problems in a complete graph and in a complete bipartite graph.

Proposition 6.4. *The degree-constrained edge k -partitioning problem P' in a complete graph is at least as difficult as the degree-constrained edge k -partitioning problem P in a complete bipartite graph.*

Proof: We are given a problem P defined by a complete bipartite graph $G = K_{X,Y}$ and values h_i^s ($i \in X$), v_j^s ($j \in Y$) for $1 \leq s \leq k$. We recall that conditions 6.1-6.3 do hold. We construct a complete graph $G' = K_{X \cup Y}$ on $X \cup Y$ by introducing in G a clique on X and a clique on Y . Let $m = |X|$ and $n = |Y|$. For each $i \in X$ we set $h_i'^1 = h_i^1 + m - 1$, $h_i'^s = h_i^s$ ($s = 2, \dots, k$) and for each $j \in Y$ we set $v_j'^2 = v_j^2 + n - 1$, $v_j'^s = v_j^s$ ($1 \leq s \leq k$, $s \neq 2$). This defines a problem P' on G' . If P has a solution S , we can derive a solution S' to P' by keeping the colors of the edges $[x_i, y_j]$ of G' , by giving color 1 to all edges $[x_r, x_t]$ and color 2 to all edges $[y_u, y_v]$. Conversely assume that P' has a solution S' in G' . Then all edges with both ends in X (resp. in Y) have color 1 (resp. color 2). In fact, suppose an edge $[x_i, x_j]$ has some color $c \neq 1$; then x_i and x_j are adjacent to $m - 2$ edges of color 1 (instead of $m - 1$) with both ends in X ; so the number of edges of color 1 going out of X will be at least $\sum_{i \in X} h_i^1 + 2 > \sum_{j \in Y} v_j^1 = \sum_{j \in Y} v_j'^1$ which is at least as large as the number of edges of color 1 which may have one or two ends in Y . This is impossible. For color 2, the same holds (interchanging the roles of X and Y). Then by keeping the colors of all edges $[x, y]$ of $K_{X \cup Y}$, we get a solution for P in $K_{X,Y}$. ■

From Proposition 6.4 and from the \mathcal{NP} -completeness of the basic image reconstruction problem for $k = 4$ [16], we obtain the following.

Corollary 6.5. *For any fixed $k \geq 4$, the degree-constrained edge k -partitioning problem in a complete graph is \mathcal{NP} -complete.*

Here we shall deal with the case where we have $k = 3$ colors. We can give an analogous statement to Proposition 6.2. We first consider the case where E^{12} is a tree.

Proposition 6.6. *In a complete graph $G = K_X$ there exist disjoint subsets E^1 and E^2 of edges such that 6.6 holds for each vertex i and for $s = 1, 2$, and E^{12} is a tree, if and only if*

$$(a) \sum_{i \in X} (h_i^1 + h_i^2) = 2(|X| - 1);$$

$$(b) \sum_{i \in X} h_i^s \leq 2(m_s - 1) \text{ for } s = 1, 2 \text{ where } m_s \text{ is the number of vertices } i \text{ with } h_i^s > 0.$$

The proof follows the same lines as the proof of Proposition 6.2 (except that we do not have to take care about the bipartite character of E^{12} when connecting different components).

The case where E^{12} is a collection of chains between odd vertices could be considered as before. If we have two odd vertices exactly then the problem amounts to finding a subset E^{12} (satisfying the degree requirements) which is a Hamiltonian chain with fixed end vertices.

We may as well consider the case where a Hamiltonian cycle has to be constructed while taking the condition 6.6 into account.

Proposition 6.7. *Given values h_i^1 and h_i^2 , satisfying $h_i^1 + h_i^2 = 2$ for each vertex i of a complete graph $G = K_X$, there are disjoint subsets E^1 and E^2 of the edge set $E(G)$ such that 6.6 holds for each vertex i and for $s = 1, 2$ and in addition E^{12} is a Hamiltonian cycle, if and only if there exists at least one vertex with $h_i^1 = h_i^2 = 1$.*

Proof: If the condition does not hold, no connected solution can be found. The sufficiency is shown by the Recoloring Lemma. The only case where it cannot be applied is when E^{12} consists of two disjoint elementary cycles which are monochromatic (one with color 1, the other one with color 2), but this is impossible from the condition. ■

6.4 A short incursion in the field of ‘oriented’ discrete tomography

In order to further generalize the previous formulations of these variations on the basic image reconstruction problem, we could consider that the underlying graph G is now oriented with arcs (x, y) instead of edges $[x, y]$. We shall assume that when two vertices x and y are linked in G , there may be several arcs (x, y) . This will be needed for constructing an initial solution.

Let us consider here the case where $G = \overrightarrow{K}_X$ is a complete symmetric oriented graph on a set X of vertices with $|X| = m$.

For each vertex i in X we are given $2k$ integers h_i^{+s}, h_i^{-s} for $s = 1, \dots, k$. We have to find a partition $\overrightarrow{E}^1, \dots, \overrightarrow{E}^k$ of the arc set $\overrightarrow{E}(G)$ such that for each color s we have

$$h_i^{+s} \text{ is the number of arcs of } \overrightarrow{E}^s \text{ leaving vertex } i \text{ in } X \quad (6.12)$$

$$h_i^{-s} \text{ is the number of arcs of } \overrightarrow{E}^s \text{ entering vertex } i \text{ in } X \quad (6.13)$$

Clearly for a solution to exist we must have

$$\sum_{s=1}^k h_i^{+s} = d_G^+(i) \quad (\text{outdegree of } i \text{ in } G) \quad (6.14)$$

$$\sum_{s=1}^k h_i^{-s} = d_G^-(i) \quad (\text{indegree of } i \text{ in } G) \quad (6.15)$$

$$\sum_{i \in X} h_i^{+s} = \sum_{i \in X} h_i^{-s} \quad (s = 1, \dots, k) \quad (6.16)$$

We assume that 6.14 - 6.16 are verified. As before we shall assume that for each color s the values h_i^{+s} and h_i^{-s} are such that there exists a subset \vec{E}^s satisfying 6.12 and 6.13. Necessary and sufficient conditions for the existence of such a subset \vec{E}^s are given in [9] (Chapter 6); to construct such a subset \vec{E}^s , we have to find a b -factor in a bipartite graph $G = (X, X', U)$ obtained by introducing for every vertex $i \in X$ a vertex $i' \in X'$ and linking every $i \in X$ to every $j' \in X'$ (with $i \neq j$) by an arc (i, j') . We set $b(i) = h_i^{+s}$ for each $i \in X$ and $b(i') = h_i^{-s}$ for every $i' \in X'$. Finding a b -factor can be done in polynomial time with network flow techniques.

As in the previous sections, we shall consider here the case of $k = 3$ colors. We assume w.l.o.g. that there is no vertex with $h_i^{+1} = h_i^{-1} = h_i^{+2} = h_i^{-2} = 0$.

Proposition 6.8. *Let $G = \vec{K}_X$ be a complete symmetric oriented graph with values h_i^{+s} and h_i^{-s} given for each vertex i in X and for colors $s = 1, 2$. There exist disjoint subsets \vec{E}^1 and \vec{E}^2 of the arc set $\vec{E}(G)$ satisfying 6.12 - 6.13 and such that $\vec{E}^{12} = \vec{E}^1 \cup \vec{E}^2$ is a tree if and only if*

$$(a) \sum_{i \in X} (h_i^{+1} + h_i^{+2}) = \sum_{i \in X} (h_i^{-1} + h_i^{-2}) = |X| - 1;$$

$$(b) \sum_{i \in X} h_i^{+s} = \sum_{i \in X} h_i^{-s} \leq m_s - 1 \quad \text{for } s = 1, 2$$

where m_s is the number of vertices i in X with $h_i^{+s} + h_i^{-s} > 0$ (i.e., vertices adjacent to at least one arc of color s).

Proof: Condition (a) is necessary for \vec{E}^{12} to be a tree. Furthermore if there is a solution, then \vec{E}^1 and \vec{E}^2 have to be forests, so (b) must hold.

Let us now show that the conditions are sufficient. By our assumptions one can find subsets \vec{E}^1 and \vec{E}^2 of $\vec{E}(G)$ satisfying 6.12 and 6.13. Notice that \vec{E}^1 and \vec{E}^2 may use parallel arcs $((x, y)_1, (x, y)_2, \dots, \text{ or } (y, x)_1, (y, x)_2, \dots)$ between pairs of vertices $x \in X, y \in X$. But since \vec{E}^{12} has to be a tree, these parallel arcs will have to be removed during the process.

Consider $\vec{E}^{12} = \vec{E}^1 \cup \vec{E}^2$; if it generates a connected graph, it is a tree from (a) and we are done.

Otherwise \vec{E}^{12} generates several connected components; at least one of them is a tree (from (a)). Now let us suppose that we can find two connected components C_1 and C_2 such that C_1 contains at least one cycle C . Take an arc (x_1, y_1) in C and assume there is in C_2 an arc (x_2, y_2) of the same color.

Replacing (x_1, y_1) and (x_2, y_2) by (x_1, y_2) and (x_2, y_1) gives a single connected component and the conditions 6.12 and 6.13 are still satisfied. No new pair of parallel arcs is created and the graph generated by \vec{E}^{12} has one less connected component. We can then apply systematically the Recoloring Lemma and the Recycling Lemma as in the bipartite case.

This can be repeated until we get a tree for \vec{E}^{12} . ■

6.5 Sufficient conditions for a solution of the image reconstruction problem with $k = 3$ colors

In this section we shall impose no requirements on the structure of E^1 and E^2 . We shall give a sufficient condition for a partition E^1, E^2, E^3 satisfying 6.4 - 6.5 to exist. So we are considering here a special case of the basic image reconstruction problem with $k = 3$ colors. We recall that its complexity status is open. The condition involves the largest degree p in $E^{12} = E^1 \cup E^2$. We shall assume that $p \geq 2$ in this section (since the case $p = 1$ is trivial).

Proposition 6.9. *In a complete bipartite graph $G = K_{X,Y}$ let $p = \max_{i \in X, j \in Y} \{h_i^1 + h_i^2, v_j^1 + v_j^2\} \geq 2$. There exists a partition E^1, E^2, E^3 of E satisfying 6.4 - 6.5 if $|E^{12}| \geq 2p(p-2) + 3$.*

Proof: By [65], we know how to construct separately E^1 and E^2 . If there is no cycle of length 2, then we are done. E^1 and E^2 are disjoint and the remaining (uncolored) edges will necessarily belong to E^3 .

Otherwise we have at least one cycle of length 2, $[x, y]_1, [x, y]_2$, where $x \in X, y \in Y, [x, y]_1 \in E^1$ and $[x, y]_2 \in E^2$. If we can find an edge $[z, t] \in E^{12}$ ($z \in X, t \in Y$) of color s such that $[x, t], [z, y] \notin E^{12}$, then by replacing $[x, y]_s$ and $[z, t]$ by $[x, t]$ and $[z, y]$ which get color s , conditions 6.4 and 6.5 are still satisfied and we have at least one less cycle of length 2. By repeating this procedure while there is a cycle of length 2, we will finish by getting 2 disjoint edge sets E^1 and E^2 satisfying 6.4 - 6.5 and thus we get a solution of our problem.

Let us now show in which case we can always find an edge $[z, t] \in E^{12}$ such that $[x, t], [z, y] \notin E^{12}$. Such an edge will be called a ‘good’ edge. Notice that x and y are considered as linked by two edges. Clearly all edges having as endvertices x or y are not ‘good’. We have at most $2(p-1)$ such edges. Furthermore, all edges $[u, v] \in E^{12}$ such that $[x, v]$ or $[u, y]$ belongs to E^{12} are not ‘good’; there are at most $2(p-2)(p-1)$ such edges. Every other edge in E^{12} not belonging to these two sets will be a ‘good’ edge. Thus if we require that E^{12} contains at least $2(p-1) + 2(p-2)(p-1) + 1 = 2p(p-2) + 3$ edges, there will always be a ‘good’ edge and hence all cycles of length 2 can be replaced by two disjoint edges. ■

We will now deal with the analogous case where $G = K_X$ is a complete graph.

Proposition 6.10. *In a complete graph $G = K_X$, let $p = \max_{i \in X}(h_i^1 + h_i^2)$, $p \geq 2$. There exists a partition E^1, E^2, E^3 of the edge set E satisfying 6.6 if $|E^{12}| \geq p^2 - 2p + 3$.*

Proof: We proceed in a similar way as in the proof of Proposition 6.9, i.e., we first construct E^1 and E^2 independently (using the construction given in [66]). If they are disjoint, then we are done. Otherwise there is at least one cycle of length 2, say $[x, y]_1, [x, y]_2$, where $[x, y]_1 \in E^1$ and $[x, y]_2 \in E^2$.

If we find an edge $[z, t] \in E^{12}$ of color s such that $[x, z], [y, t] \notin E^{12}$ or $[x, t], [y, z] \notin E^{12}$, then we can replace $[z, t]$ and $[x, y]_s$ by one of these pairs of edges which will get color s and condition 6.6 will still be satisfied. Repeating this procedure will necessarily lead to a solution.

Now we will show a sufficient condition for such an edge $[z, t]$, called ‘good’ edge, to exist. Clearly all edges incident to x or y in E^{12} are not ‘good’. We have at most $2(p - 1)$ such edges. Let q denote the number of vertices w which are common neighbors of x and y in E^{12} . Then all edges incident to these vertices are not ‘good’ either. We have at most $q(p - 2)$ of them different from edges $[x, w], [y, w]$. Finally each edge $[u, v]$ joining two neighbors of x (resp. y) which are not neighbors of y (resp. x) is not ‘good’. We have at most $(p - 2 - q)(p - 3 - q)$ such edges. It is easy to see that every other edge will be a ‘good’ one. Thus if we require that $|E^{12}| \geq 2(p - 1) + q(p - 2) + (p - 2 - q)(p - 3 - q) + 1$, then we can always find a ‘good’ edge and hence replace each cycle of length 2. If we consider the extreme cases where $q = 0$ and $q = p - 2$, we find that $|E^{12}| \geq \max(p^2 - 2p + 3, p^2 - 3p + 5)$ and thus $|E^{12}| \geq p^2 - 2p + 3$ since $p \geq 2$. ■

6.6 Cases where each one of E^1, E^2 is structured

We shall now examine additional cases where the number k of colors is $k = 3$ and the subsets E^1 and E^2 have a given structure. The first situation will be the following. Each one of E^1 and E^2 is a Hamiltonian chain in the complete bipartite graph $K_{X,Y}$ associated with the array $A = (a_{ij})$. For this graph to have Hamiltonian chains we shall assume $|X| = |Y|$ (notice that we could have $|X| \leq |Y| \leq |X| + 1$ but for simplicity we will limit our study to the case where $|X| = |Y|$) and each chain has an endvertex in X and the other one in Y . Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ and let $a \in X$ and $b \in Y$ be the endvertices of the Hamiltonian chain forming E^1 ; this means that we have

$$h^1(x_i) = \begin{cases} 1 & \text{if } x_i = a; \\ 2 & \text{if } x_i \neq a; \end{cases}$$

and

$$v^1(y_i) = \begin{cases} 1 & \text{if } y_i = b; \\ 2 & \text{if } y_i \neq b. \end{cases}$$

Let $u \in X$ and $v \in Y$ be the endvertices of the Hamiltonian chain E^2 ; we will have similarly

$$h^2(x_i) = \begin{cases} 1 & \text{if } x_i = u; \\ 2 & \text{if } x_i \neq u; \end{cases}$$

and

$$v^2(y_i) = \begin{cases} 1 & \text{if } y_i = v; \\ 2 & \text{if } y_i \neq v. \end{cases}$$

Proposition 6.11. *In $G = K_{X,Y}$ (with $|X| = |Y| = n$) there exist two disjoint Hamiltonian chains E^1 (with arbitrary endvertices a and b) and E^2 (with arbitrary endvertices u and v) if and only if $n \geq 5$.*

Proof: For $n \leq 4$ one cannot find two disjoint Hamiltonian chains in $K_{n,n}$ for arbitrary endvertices. For instance one cannot find two disjoint Hamiltonian chains with $a = u$ and $b = v$. We shall therefore assume now that $n \geq 5$. Let x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n be a numbering of the vertices in X and in Y such that $x_1 = a$ and $y_1 = b$. We construct E^1 by taking edges $[x_i, y_{i+1}], [x_{i+1}, y_i]$, for $i = 1, \dots, n - 1$, and $[x_n, y_n]$. We have several cases to consider for E^2 .

(a) $u = a, v = b$

For $n \geq 5$ we construct E^2 as follows (see Figure 6.1). We build the sequence of indices of vertices which will be visited by E^2 by taking first the odd indices in increasing order followed by the even indices in increasing order but where we just interchange 4 with the largest even index $2p$; the sequence \vec{C} obtained in this way is then completed by the same sequence \overleftarrow{C} in reverse order; then we assign labels x and y alternately to all terms of the sequence $\vec{C} \oplus \overleftarrow{C}$; we get thus $x_1, y_3, \dots, x_3, y_1$. This gives a Hamiltonian chain E^2 with endvertices x_1 and y_1 which is disjoint from E^1 since E^2 contains neither edges $[x_i, y_j]$ with $|i - j| = 1$ nor $[x_n, y_n]$.

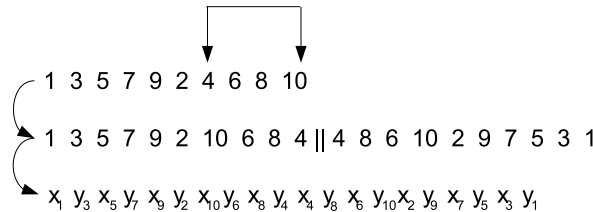


Figure 6.1: Construction of E^2 in the case of $n = 10$.

(b) $u = a, v \neq b$

Notice that v is a fixed arbitrary vertex of Y with $v \neq y_1$. We can w.l.o.g. assume that from the beginning our vertices have been numbered in such a way that $v = y_3$. Now replacing $[x_1, y_3]$ by $[x_1, y_1]$ in the E^2 constructed in a) gives a new Hamiltonian chain E^2 disjoint from E^1 and with endvertices u and v .

(c) $u \neq a, v \neq b$

Again u and v are fixed arbitrary vertices of X and Y respectively. We can w.l.o.g. assume that from the beginning our vertices have been numbered in such a way that $u = x_4$ and $v = y_4$. We obtain E^2 by replacing $[x_4, y_4]$ by $[x_1, y_1]$ in the E^2 constructed in a). Clearly E^2 is a Hamiltonian chain disjoint from E^1 .

■

Remark 6.12. *One can easily verify that Proposition 6.11 can be extended to the case of a complete graph $G = K_X$ ($|X| = m$) with $m \geq 4$ if $a \neq u$ and $b \neq v$, $m \geq 5$ if $a = u$ and $b \neq v$, $m \geq 6$ if $a = u$ and $b = v$.*

Consider now the case where for each color $s = 1, 2$ we have $h_i^s, v_j^s \in \{0, 2\}$ for all i, j . $V(E^s)$ will be the set of vertices with $h_i^s = 2$ or $v_j^s = 2$ for $s = 1, 2$. The problem consists then in finding two disjoint cycles E^1 and E^2 through specified vertex sets $V(E^1)$ and $V(E^2)$. W.l.o.g. we can assume that 6.9 holds: $V(G) = X \cup Y = V(E^1) \cup V(E^2)$.

The reconstruction problem where both E^1 and E^2 are collections of vertex disjoint cycles was studied in [18] under the name $RPB(m, n, p = 2)$ (see also [11, 76]). It was shown that a solution could be constructed if and only if one did not have one of four pathological cases:

- (a) $\sum_{i \in X} h_i^1 = 4 = \sum_{i \in X} h_i^2, |X| \leq 3, |Y| \leq 3;$
- (b) $\sum_{i \in X} h_i^1 = 4, \sum_{i \in X} h_i^2 = 6, |X| = 3, |Y| \leq 4;$
- (c) $\sum_{i \in X} h_i^1 = 6 = \sum_{i \in X} h_i^2, |X| = 3, |Y| \leq 5;$
- (d) $\sum_{i \in X} h_i^1 = 6, \sum_{i \in X} h_i^2 = 8, |X| = 4 = |Y|.$

For more results on disjoint cycles in graphs, we refer the reader to [55] where the case of Hamiltonian cycles is considered.

Goddyn and Stacho give in [37] the following theorem concerning general graphs.

Theorem 6.13. [37] *Let $G = (V, E)$ be a finite undirected simple graph of order m , let $W \subseteq V$, $|W| \geq 3$, and let k be a positive integer. Suppose that $G[W]$ is $2k$ -connected, and that*

$$\max(d_G(u), d_G(v)) \geq \frac{m}{2} + 2(k - 1) \tag{6.17}$$

for every $u, v \in W$ such that $\text{dist}_{G[W]}(u, v) = 2$. Then G contains k pairwise edge-disjoint cycles E^1, \dots, E^k such that $W \subseteq V(E^i)$, $1 \leq i \leq k$.

(Here, $G[W]$ is the subgraph induced by W , and $\text{dist}_G(u, v)$ is the distance from u to v in G .)

Notice that for $k \geq 2$ condition 6.17 cannot be verified in the case of bipartite graphs. For complete bipartite graphs we give the following.

Proposition 6.14. *Let $K_{X,Y}$ be a complete bipartite graph. There exist two edge-disjoint cycles E^1 and E^2 through specified vertex sets $V(E^1)$ and $V(E^2)$ if and only if we are not in one of the four pathological cases and we do not have the forbidden configuration F given in Figure 6.2.*

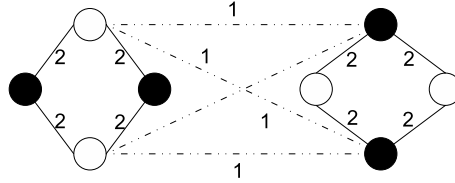


Figure 6.2: Configuration F admitting no connected solution. The black vertices belong to X and the white ones to Y .

Proof: First, if $\sum_{i \in X} h_i^2 \geq 10$, then, whenever 6.1–6.3 are satisfied, there always exists a ‘connected’ solution (i.e., two disjoint cycles E^1 and E^2) satisfying 6.4 and 6.5. Indeed, remove a cycle of length $\sum_{i \in X} h_i^1$ (the cycle E^1 through $V(E^1)$ corresponding to edges of color 1) from $K_{X,Y}$. Then, if we consider any p by p induced subgraph of the resulting graph (and so, in particular, the one induced by the $\sum_{i \in X} h_i^2 = 2p$ vertices adjacent to edges of color 2), it has a minimum degree of $p - 2$. Since $\sum_{i \in X} h_i^2 \geq 10$, we have $p \geq 5$ and so $p - 2 \geq \frac{p+1}{2}$. Thus, by [4] (Chapter 7, Section 3), there exists a Hamiltonian cycle in this (sub)graph which will correspond to E^2 . This implies that we can obtain two disjoint cycles (one for color 1 and one for color 2) respecting the given projections.

Second, if $\sum_{i \in X} h_i^1 = \sum_i h_i^2 = 8$, then, as previously, there always exists a connected solution whenever 6.1–6.3 are satisfied. Take any cycle E^1 on $V(E^1)$. Say $E^1 = \{[x_1, y_1], [y_1, x_2], \dots, [x_4, y_4], [y_4, x_1]\}$. Consider the cycle $C = \{[x'_1, y'_1], [y'_1, x'_2], \dots, [x'_4, y'_4], [y'_4, x'_1]\}$, where $x'_i = x_i$ if $h^2(x_i) = 2$ and $x'_i = z_i$ if $h^2(x_i) = 0$, where $z_i \neq x_j$, $j = 1, 2, 3, 4$, is some vertex in X with $h^2(z_i) = 2$ as well as $y'_i = y_i$ if $v^2(y_i) = 2$ and $y'_i = t_i$ if $v^2(y_i) = 0$, where $t_i \neq y_j$, $j = 1, 2, 3, 4$, is some vertex in Y with $v^2(t_i) = 2$. We construct E^2 by linking each vertex v in C to the two vertices in C which are at distance three of v .

Finally, let us deal with the case where $\sum_{i \in X} h_i^1 \leq 6$ and $\sum_{i \in X} h_i^2 \leq 8$. First, it is easy to see that the example given in Figure 6.2 (where $\sum_{i \in X} h_i^1 = 4$ and $\sum_{i \in X} h_i^2 = 8$) does not have an equivalent connected solution. If $\sum_{i \in X} h_i^1 \leq 6$ and $\sum_{i \in X} h_i^2 \leq 6$, then, if we are not in one of the four pathological cases, there exists a solution and this solution is necessarily

connected (since there can be no cycle of length three or less). The last case to consider is when $\sum_{i \in X} h_i^2 = 8$ and $\sum_{i \in X} h_i^1 \leq 6$. Assume we are not in one of the four pathological cases (and thus there exist two disjoint subsets of edges E'^1 and E'^2 satisfying 6.4 and 6.5 (E'^1 necessarily forming a single cycle)) and that the subgraph induced by $V(E'^1) \cup V(E'^2)$ is not the one in Figure 6.2. If E'^2 consists of one C_8 , we are done. Otherwise (E'^2 consists of two C_4), if there exists in the solution a C_4 with edges of color 2,3,2 and 3, then color 2 and color 3 can be interchanged in the C_4 . This provides an equivalent solution in which the edges of color 2 form a C_8 . If such a cycle does not exist, then anyway there exists at least one edge of color 3 between the vertices of the two C_4 of color 2 (since otherwise there are 8 edges of color 1 between these vertices, and this contradicts $\sum_{i \in X} h_i^1 \leq 6$). This implies that there exist four edges of color 1 between these vertices (since otherwise there is a C_4 with edges of color 2,3,2 and 3), which form a C_4 . Therefore, this C_4 is *the* cycle of color 1. Hence, we obtain the graph in Figure 6.2, which is a contradiction. ■

Consider now the case where both E^1 and E^2 are cycles in a simple complete graph $G = K_X$ with $|X| = m$, i.e., $h_i^s = 0$ or 2 for $s = 1, 2$. We shall assume that $V(E^1) \cup V(E^2) = X$ and $|V(E^1)|, |V(E^2)| \geq 3$. We have the following.

Proposition 6.15. *In $G = K_X$ with $|X| = m$, one can find two edge-disjoint cycles E^1 on $V(E^1)$ and E^2 on $V(E^2)$ if and only if we are not in the following cases.*

- (a) $m \leq 4$;
- (b) $3 \leq |V(E^1) \cap V(E^2)| \leq 4$ and $m = 5$.

Proof: It is easy to see that the conditions are necessary. If we are in case (a) or (b), one can easily check that E^1 and E^2 cannot be disjoint, by enumerating all possible configurations (case (a) or (b) and different lengths of the cycles (3, 4, or 5)).

Consider now the case where $m \geq 5$.

From Theorem 6.13 we deduce that the proposition is true for $|W| = |V(E^1) \cap V(E^2)| \geq 5$. Indeed, in a complete graph there are no two vertices at distance 2 and W is 4-connected if $|W| \geq 5$. Thus, the conditions of Theorem 6.13 are verified and the graph contains 2 edge-disjoint cycles through $V(E^1)$ and $V(E^2)$.

If $|W| < 5$, W is not 4-connected and the conditions do not hold anymore. Let us now study the different cases.

If $|V(E^1) \cap V(E^2)| \leq 1$, then E^1 and E^2 will be edge disjoint.

If $V(E^1) \cap V(E^2) = \{a, b\}$, then w.l.o.g. $|V(E^1) \setminus (V(E^1) \cap V(E^2))| \geq 2$, say $V(E^1) \setminus (V(E^1) \cap V(E^2)) = \{c, d, \dots, z\}$. We construct a cycle $E^1 = a, c, b, d, \dots, z, a$ and for E^2 a cycle through $[a, b]$ followed by a chain between a and b in $V(E^2) \setminus (V(E^2) \cap V(E^1))$. These subsets E^1 and E^2 will be disjoint.

If $|V(E^1) \cap V(E^2)| = 3$ or 4 and $m \geq 6$, it is easy to obtain two disjoint cycles. We give the construction for $|V(E^1) \cap V(E^2)| = 4$ (the case $|V(E^1) \cap V(E^2)| = 3$ can be treated

similarly). Let $V(E^1) \cap V(E^2) = \{a, b, c, d\}$. Then $|V(E^1) - V(E^2)| + |V(E^2) - V(E^1)| \geq 2$. If $e \in V(E^1) - V(E^2)$ and $f \in V(E^2) - V(E^1)$, we can take for E^1 the cycle beginning with vertices a, b, c, d, e, \dots followed by the remaining vertices of $V(E^1) - V(E^2)$. For E^2 we can take the cycle beginning with vertices c, a, d, b, f, \dots followed by the remaining vertices of $V(E^2) - V(E^1)$. If f does not exist, i.e., $V(E^2) \subseteq V(E^1)$ then we have $V(E^1) - V(E^2) = \{e, g, \dots\}$. In this case we take for E^1 the cycle beginning with vertices a, b, e, c, d, g, \dots followed by the remaining vertices of $V(E^1) - V(E^2)$. For E^2 , we take the cycle (a, d, b, c, a) . Clearly E^1 and E^2 will be disjoint. ■

We have considered here the problem of constructing in a complete (bipartite) graph two disjoint subsets of edges satisfying some requirements on their degrees at every vertex. Since the given values h_i^s and v_j^s determine the cardinalities $|E^1|$ and $|E^2|$, we have in fact to find if there exist two disjoint subsets of edges with given cardinalities which satisfy some additional requirements (on their degrees). This problem is related to the following \mathcal{NP} -complete problem (see [35]). Given a bipartite graph G and two integers $p > q > 0$, does G contain two edge disjoint matchings M_p and M_q with $|M_p| = p, |M_q| = q$? In case the values h_i^s and v_j^s are 0 or 1, then E^1 and E^2 are matchings of a fixed size. But we know which vertices have to be saturated by E^1 and/or E^2 and the graph G is a complete bipartite graph. This situation has been studied in [18] under the name $RPU(m, n, p)$.

6.7 Conclusion

We have investigated some graph theoretical problems related to the basic image reconstruction problem in discrete tomography. We have exhibited a solvable case of the basic image reconstruction problem with $k = 3$ colors. The complexity of the related problem in a complete graph has been settled for a fixed $k \geq 4$.

We imposed the structure of the graph formed by the union of two colors. Here having a tree allowed us to find solutions whenever they existed. The choice was adequate since it eliminated the cycles that were introduced by the parallel edges or arcs needed in the model (the presence of parallel edges in E^{12} would have meant that the corresponding entries of the array A received several colors!). In fact we have imposed constraints on the cardinalities of E^1, E^2 and/or E^{12} and it is worth observing that if we introduce some requirements on $|E^1|$ (for instance $|E^1| \leq f(m, n)$ where f is a linear function of the size of the array $A = (a_{ij})$), this additional piece of information does not simplify the problem in the following sense. One may transform any reconstruction problem P with $k = 3$ colors into a larger problem P' with $k = 3$ colors where the first color class will satisfy a requirement of the form $|E^1| \leq f(m', n')$ and P' will have a solution if and only if P has one. This can be seen easily by embedding array A in a corner of a larger $(m' \times n')$ array A' where we impose color 3 to all entries of A' outside of A .

It would be interesting to examine other structures for the graph associated with the union

of several colors.

Finally we mention the case where each one of E^1 and E^2 is a spanning tree; this problem seems to be of interest. As far as we know problems consisting of packing some special types of graphs, like trees, have not been explored intensively when there are requirements on the degrees of the vertices.

Conclusion

We examined in this thesis some variations of coloring problems which arise from scheduling and discrete tomography. We obtained several results concerning the computational complexity of these problems, i.e., we considered special cases and either we developed polynomial time algorithms or showed that even in these restricted cases the problems remain hard to solve. This work suggests ideas for further research concerning these topics.

In Chapter 2 we considered two coloring problems in mixed graphs arising from scheduling. We gave some upper bounds on the strong and the weak mixed chromatic number and presented some complexity results. For both problems, it should be interesting to determine their complexity in other classes of graphs, e.g. in mixed planar cubic bipartite graphs. But also mixed graphs $G_M = (V, U, E)$ for which the directed partial graph $G_M^o = (V_o, U, \emptyset)$ has a specific structure (e.g. collection of disjoint paths, arborescence) should be considered. These kinds of graphs are maybe the most promising for finding more polynomially solvable cases. Furthermore, it should be interesting to find more lower and upper bounds on the weak mixed chromatic number.

We examined in Chapter 3 the following problem: in an undirected graph $G = (V, E)$ we want to characterize a minimum set R of edges for which maximum matchings M can be found with specific values of $p = |M \cap R|$. In general our problem requires the determination of a shortest alternating cycle with respect to some maximum matching (not given). As far as we know the complexity status of this problem is still open. For 3-regular bipartite graphs $G = (X, Y, E)$ with $|X| = |Y| = n \geq 2(p - 1)$, where $p \leq 4$ is an integer, we determined a minimum set R which is \mathcal{P} -feasible for $\mathcal{P} = \{0, 1, \dots, p\}$. The construction does not seem to hold for $p \geq 5$, thus it would be interesting to find another construction or to adapt the first one. Also the case of general odd cacti would be interesting to analyze since these graphs are exactly the IP-perfect graphs.

In Chapters 4, 5 and 6 we examined some coloring (resp. partitioning) problems which arise from discrete tomography. These problems are generalizations or variations of the basic image reconstruction problem.

We considered in Chapter 4 a generalization of the vertex coloring problem associated with the basic image reconstruction problem. Here the vertices of the graph are covered by chains and for each chain the number of occurrences of each color is given. We then want to find out whether there exists a coloring of the vertex set respecting these occurrences. We gave several complexity results for the case of arbitrary colorings (two adjacent vertices may have the same color) as well as for the case of proper colorings.

Most of the results concerned graphs for which we did some restrictive hypothesis. Of course considering more general cases would be interesting but our results show that the problems become difficult even in very simple cases.

Instead of considering chains, we considered in Chapter 5 some generalized neighborhoods. We obtained for the 2-neighborhood a polynomial time algorithm for quatrees (i.e. trees where all internal vertices have degree at least 4). It does not seem easy to adapt our algorithm to general trees. So it would be interesting to try another approach or detect some more special cases of trees which are polynomially solvable. Considering quatrees allowed us to obtain a unique solution, if it existed. In the case of closed neighborhood this does not seem to be true anymore and hence it will be difficult to adapt our algorithm to this case in trees or even quatrees.

We also examined the bounded version of these problems, i.e., instead of the exact number of occurrences of each color we are given upper bounds on these occurrences. Some complexity results were given for the case of open neighborhood and three or four colors. It would be interesting to analyze the bounded version in the case of the 2-neighborhood for trees or quatrees.

Finally in Chapter 6 we considered the edge partitioning version of the basic image reconstruction problem. We presented a new solvable case for $k = 3$ colors. Then we examined some variations where the union of the two subsets E^1 and E^2 have to satisfy some additional constraints. One should consider other cases where some of the sets E^s are imposed to have a certain structure or where they form a special graph. One may also consider the case of $k \geq 3$ colors where the union of more than two subsets must form a special graph. An open question is whether the reconstruction problems in complete graphs and complete bipartite graphs are equivalent from a computational complexity point of view. In other words, can we transform the reconstruction problem in a complete graph into an equivalent reconstruction problem in a complete bipartite graph?

Lots of open questions exist and further research is needed to detect more solvable cases or to strengthen existing complexity results for these problems. Probably the most important open question which remains is the complexity of the basic image reconstruction problem for $k = 3$ colors. Since there exist so far only very few polynomial solvable cases for this problem, this could also be a direction of further research. Concerning the mixed

graph coloring problems, there are some cases left where the complexity is unknown and it would be interesting to identify more classes of graphs where the problems are polynomially solvable.

Bibliography

- [1] R.K. Ahuja, T.L. Magnanti, and J. Orlin. *Networks flows*. Prentice-Hall, 1993.
- [2] G.V. Andreev, Y.N. Sotskov, and F. Werner. Branch and bound method for mixed graph coloring and scheduling. In *Proceedings of the 16th International Conference on CAD/CAM, Robotics and Factories of the Future, CARS and FOF 1 2000*, pages 1–8. Trinidad and Tobago, 2000.
- [3] B. Aspvall, M.F. Plass, and R. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8:121–123, 1979.
- [4] A.S. Asratian, T.M.J. Denley, and R. Häggkvist. *Bipartite graphs and their applications*. Cambridge University Press, Cambridge, 1998.
- [5] B.S. Baker and E.G. Coffman. Mutual exclusion scheduling. *Theoretical Computer Science*, 162:225–243, 1996.
- [6] C. Bentz, M.-C. Costa, D.de Werra, C. Picouleau, and B. Ries. On a graph coloring problem arising from discrete tomography. *Networks*, to appear.
- [7] C. Bentz, M.-C. Costa, D.de Werra, C. Picouleau, and B. Ries. Degree-constrained edge partitioning in graphs arising from discrete tomography. Technical Report ORWP 07/03, submitted for publication, EPFL, 2007.
- [8] C. Berge. Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind. *Technical Report Wiss. Zeitung 114*, Martin Luther University, Halle-Wittenberg, 1961.
- [9] C. Berge. *Graphes*. Gauthier-Villars, Paris, 1983.
- [10] G. Bitton and M. Munz. Le cactus. *Private communication*, Paris, December 14 2006.
- [11] B. Bollobás and G. Brightwell. Cycles through specified vertices. *Combinatorica*, 13:147–155, 1993.
- [12] T. Calamoneri. The L(h,k)-labelling problem: a survey and annotated bibliography. *Theoretical Computer Science*, 49 (5):585–608, 2007.

- [13] R. Chandrasekaran, S.N. Kaboadi, and K.G. Murty. Some NP-complete problems in linear programming. *Operations Research Letters*, 1:101–104, 1982.
- [14] B.-L. Chen and K.-W. Lih. Equitable coloring of trees. *Journal of Combinatorial Theory, Series B*, 61 (1):83–87, 1994.
- [15] M. Chlebik and J. Chlebikova. Hard coloring problems in low degree planar bipartite graphs. *Discrete Applied Mathematics*, 154:1960–1965, 2006.
- [16] M. Chrobak and C. Dürr. Reconstructing polyatomic structures from X-rays: NP-completeness proof for three atoms. *Theoretical Computer Science*, 259(1):81–98, 2002.
- [17] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York. Association for Computing Machinery, 1971.
- [18] M.-C. Costa, D. de Werra, and C. Picouleau. Using graphs for some discrete tomography problems. *Discrete Applied Mathematics*, 154(1):35–46, 2006.
- [19] M.-C. Costa, D. de Werra, C. Picouleau, and D. Schindl. A solvable case of image reconstruction in discrete tomography. *Discrete Applied Mathematics*, 148:240–245, 2005.
- [20] M.-C. Costa, D. de Werra, C. Picouleau, and B. Ries. Bicolored matchings in some classes of graphs. *Graphs and Combinatorics*, 23:47–60, 2007.
- [21] M.-C. Costa, D. de Werra, C. Picouleau, and B. Ries. Graph coloring with constraints on the neighborhoods. Technical Report ORWP 06/08, submitted for publication, EPFL, 2007.
- [22] L. Cowen, W. Goddard, and C.E. Jesurum. Defective coloring revisited. *Journal of Graph Theory*, 23(3):205–219, 1997.
- [23] D. P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics*, 30:289–293, 1980.
- [24] D. de Werra. On line-perfect graphs. *Mathematical programming*, 15:236–238, 1978.
- [25] D. de Werra and J. Erschler. Open shop scheduling with some additional constraints. *Graphs and Combinatorics*, 12:81–93, 1996.
- [26] D. de Werra, A. Hertz, D. Kobler, and N.V.R. Mahadev. Feasible edge colorings of trees with cardinality constraints. *Discrete Mathematics*, 222:61–72, 2000.
- [27] D. de Werra, N.V.R. Mahadev, and U. Peled. Edge-chromatic scheduling with simultaneity constraints. *SIAM Journal on Discrete Mathematics*, 6 (4):631–641, 1993.
- [28] J. Edmonds. Matching and a polyhedron with 0,1 vertices. *Journal of Research National Bureau of Standards Section B*, 69:125–130, 1965.

- [29] J. Edmonds. Path, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [30] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Opera Omnia*, 7:128–140, 1736.
- [31] U. Feige, E. Okek, and U. Wieder. Approximating maximum edge coloring in multi-graphs. Technical report, Weizmann Institute, 2003.
- [32] G. Fertin, A. Raspaud, and B. Reed. Star coloring of graphs. *Journal of Graph Theory*, 47(3):163–182, 1997.
- [33] H. Furmańczyk, A. Kosowski, and P. Zyliński. Mixed graph coloring: Bipartite graphs, trees, and series-parallel graphs. *Manuscript*, 2006.
- [34] H.N. Gabor and R.E. Tarjan. Efficient algorithms for a family of matroid intersection problems. *Journal of Algorithms*, 5:80–131, 1984.
- [35] H. Gabow, T. Nishizeki, O. Kariv, D. Leven, and O. Tereda. Algorithms for edge-coloring. *Unpublished manuscript, University of Colorado*, 1983.
- [36] M. Garey and D.S. Johnson. *Computer and Intractability*. Freeman, San Francisco, 1979.
- [37] L. Goddyn and L. Stacho. Edge disjoint cycles through specified vertices. *Journal of Graph Theory*, 50:246–260, 2006.
- [38] J.L. Gross and J. Yell. *Handbook of graph theory*. CRC Press, London, 2004.
- [39] P. Hansen and D. de Werra. Nesticity, Dimacs series. *Discrete Mathematics and Theoretical Computer Science*, 37:225–232, 1997.
- [40] P. Hansen, A. Hertz, and J. Kuplinsky. Bounded vertex colorings of graphs. *Discrete Mathematics*, 111:305–312, 1993.
- [41] P. Hansen, J. Kuplinsky, and D. de Werra. Mixed graph colorings. *Mathematical Methods of Operational Research*, 45:145–160, 1997.
- [42] D. Hartvigsen. *Extensions of matching theory*. PhD thesis, Carnegie-Mellon University, 1984.
- [43] I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.
- [44] M. Hujter and ZS. Tuza. Pre-coloring extension in graphs classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62(1):1–11, 1993.
- [45] K. Jansen and P. Scheffler. Generalized coloring for tree-like graphs. In *18th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'92)*, pages 50–59. Lecture Notes in Computer Science 657 (ed. E. W. Mayr) 1993.

- [46] M. Jarvis and B. Zhou. Bounded vertex coloring of trees. *Discrete Mathematics*, 232:145–151, 2001.
- [47] T. Jensen and B. Toft. *Graph Coloring Problems*. John Wiley & Sons, New York, 1995.
- [48] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- [49] A.V. Karzanov. Maximum matching of given weight in complete and complete bipartite graphs. *Kibernetika*, 1:7–11, 1987.
- [50] L.I. Klimova and Y.N. Sotskov. Coloring of vertices of a mixed graph. *Algorithms for Solving Optimization Problems, Institute of Engineering Cybernetics, Minsk, Belarus (in Russian)*, pages 90–96, 1993.
- [51] J. Kratochvil. Precoloring extension with fixed color bound. *Acta Mathematica Universitatis Comenianae*, 63(1):139–153, 1994.
- [52] A. Kuba and G.T. Hermann. *Discrete tomography: Foundations, Algorithms and Applications*. Birkhäuser, 1990.
- [53] A. Kuba and G.T. Hermann. *Advances in Discrete Tomography and its Applications*. Birkhäuser, 2007.
- [54] G.M. Levin. On the evaluation of chromatic characteristics of a mixed graph. *Vesti Akademii Navuk BSSR, Seriya Fiz.-Mat Navuk (in Russian)*, 1:17–20, 1982.
- [55] G. Li. Edge disjoint Hamilton cycles in graphs. *Journal of Graph Theory*, 35:8–20, 2000.
- [56] L. Lovasz and M.D. Plummer. Matching theory. *Annals of Discrete Mathematics*, 29, 1986.
- [57] D. Marx. Graph coloring problems and their application in scheduling. *Periodica Polytechnica Ser. El. Eng.*, 48(1-2):5–10, 2004.
- [58] D. Marx. NP-completeness of list-coloring and precoloring extension on the edges of planar graphs. *Journal of Graph Theory*, 49(4):312–324, 2005.
- [59] D.W. Matula and L.L. Beck. Smallest last ordering and clustering and graph coloring algorithms. *Journal of the Association for Computing Machinery*, 30(3):417–427, 1983.
- [60] C. Moore and J.M. Robson. Hard tiling problems with simple tiles. *Discrete and Computational Geometry*, 26:573–590, 2001.
- [61] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

- [62] B. Ries. Coloring some classes of mixed graphs. *Discrete Applied Mathematics*, 155:1–6, 2007.
- [63] B. Ries and D. de Werra. On two coloring problems in mixed graphs. *European Journal of Combinatorics*, to appear.
- [64] G. Rote. Private communication.
- [65] H.J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canadian Journal of Mathematics*, 9:371–377, 1957.
- [66] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, 2003.
- [67] Y.N. Sotskov. Scheduling via mixed graph coloring. In *Operations Research Proceedings held at the Otto-von-Guericke University Magdeburg, September 1-3 1999*, pages 414–418, Magdeburg, Germany. Springer Verlag, 2000.
- [68] Y.N. Sotskov, A. Dolgui, and F. Werner. Mixed graph coloring via unit-time job-shop. *International Journal of Mathematical Algorithms*, 2:289–323, 2001.
- [69] Y.N. Sotskov and V.S. Tanaev. Chromatic polynomial of a mixed graph. *Vesti Akademii Navuk BSSR, Seriya Fiz.-Mat Navuk (in Russian)*, 6:20–23, 1976.
- [70] Y.N. Sotskov, F. Werner, and V.S. Tanaev. Scheduling problems and mixed graph colorings. *Optimization*, 51:597–624, 2002.
- [71] L.E. Trotter. Line-perfect graphs. *Mathematical Programming*, 12:255–259, 1977.
- [72] P. van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 58:297–326, 1992.
- [73] V.V. Vazirani. *Approximation algorithms*. Springer-Verlag, Berlin, 2001.
- [74] V.G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz. (in Russian)*, 3:25–30, 1964.
- [75] T. Yi, K.G. Murty, and C. Spera. Matchings in colored bipartite networks. *Discrete Applied Mathematics*, 121:261–277, 2002.
- [76] F. Zeffirelli. Les cycles franciscains de Giotto. *Private communication*, 14mn, B.1967.
- [77] N. Zufferey. *Heuristiques pour les problèmes de la coloration des sommets d’un graphe et d’affectation de fréquences avec polarités*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, 2002.

Index

$\chi_M(G_M)$, 14, 22, 34–36

$\gamma_M(G_M)$, 13, 21, 23–25

k -coloring, 11

bipartite graph, 11, 42, 44, 79, 80

cactus, 49, 51

combinatorial optimization, 1

complete bipartite graph, 2, 18, 92, 93, 97

discrete tomography, 2, 3, 16, 53, 77, 91

graph theory, 1, 2, 7

grid graph, 2, 18, 54

matching, 2, 9, 15, 42

maximum matching, 15, 42, 44

mixed graph, 2, 8, 12, 21

partition, 12, 15, 18, 53, 78, 93

perfect matching, 43

scheduling, 2, 21

strong mixed graph coloring, 13, 21, 23

tree, 10, 94, 98

weak mixed graph coloring, 13, 22, 34

Curriculum Vitae

Name	Bernard Ries
Nationality	Luxembourgish
Date of birth	June 12, 1979
Marital Status	Single
Address	Ch. de Rionza 19 CH - 1020 Renens

Education

2004–2007	PhD student of Prof. Dominique de Werra in Operations Research at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
2002–2004	Undergraduate studies, Department of Mathematics, EPFL, Switzerland
2001–2002	Undergraduate studies, Department of Mathematics and Statistics, Université de Montréal, Canada
1999–2001	Undergraduate studies, Department of Mathematics, EPFL, Switzerland
1991–1998	Highschool studies at Lycée de Garçons de Luxembourg, G.-D. of Luxembourg

Languages

Luxembourgish	Native language
French	fluently spoken and written
English	fluently spoken and written
German	fluently spoken and written

Professional Experience

2004–2007	Teaching assistant in ROSE research group at EPFL: Discrete Mathematics, Operations Research, Graphs and Networks, Mathematical Coding
2005	Teaching Graph Theory to high school professors of mathematics, Switzerland

Conferences and Talks

- 2007 *Invited talk at Gdansk Technical University*, ‘Graphs and discrete tomography’, Gdansk
- 2007 *Fifth Joint Operations Research Days*, ‘Mixed Graph Coloring’, Lausanne
- 2007 *Sixth International Symposium on Graphs and Optimisation*, ‘Degree-constrained edge partitioning in graphs arising from discrete tomography’, Cademario
- 2007 *Invited talk at Universität der Bundeswehr*, ‘Degree-constrained edge partitioning in graphs arising from discrete tomography’, Munich
- 2007 *1st Canadian Discrete and Algorithmic Mathematics Conference*, ‘Mixed Graph Coloring’ (Invited talk in session ‘Games on Graphs’), Banff
- 2007 *Conférence conjointe Francoro V/ROADEF*, ‘Reconstruction de la coloration d’un graphe à partir des projections des voisinages’, Grenoble
- 2006 *Fourth Joint Operations Research Days*, ‘Graph coloring with cardinality constraints on the neighborhoods’, Lausanne
- 2006 *Fifth International Symposium on Graphs and Optimisation*, ‘Graph coloring with cardinality constraints on the neighborhoods’, Leukerbad
- 2006 *Sixth Czech-Slovak International Colloquium on Combinatorics, Graph Theory, Algorithms and Applications*, ‘Coloring some classes of mixed graphs’, Prague
- 2006 *3e cycle romand de recherche opérationnelle*, ‘On mixed graph coloring’, Zinal
- 2005 *7th International Colloquium on Graph Theory*, ‘Bicolored matchings in some classes of graphs’, Hyères

Miscellaneous

- 2007 Co-organizer of the Sixth International Symposium on Graphs and Optimisation in Cademario
- 2006 Co-organizer of the Fifth International Symposium on Graphs and Optimisation in Leukerbad
- 2005 - 2008 Member of the executive committee of the Swiss OR Society
- 2004 - present Member of the Swiss OR Society