Graph problems motivated by (low and high) resolution models of large protein assemblies

Viet-Ha Nguyen

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Problèmes de graphes motivés par des modèles basse et haute résolution de grands assemblages de protéines

Thi Viet Ha NGUYEN

I3S, Inria Sophia Antipolis

Présentée en vue de l'obtention du grade de **docteur en informatique** d'**Université Côte d'Azur**

Dirigée par : **Frédéric Havet**
Co-encadrée par : **Dorian Mazauric**
Soutenue le : **13/12/2021**

Devant le jury, composé de :
Etienne Birmelé, PR, IRMA (rapporteur)
Guillaume Fertin, PR, LS2N (rapporteur)
Christophe Paul, DR, CNRS, LIRMM
Blerina Sinaimeri, Ass. Prof., LUISS (Italie)
Stéphane Vialette, DR, CNRS, LIGM
Frédéric Havet, DR, CNRS, I3S
Dorian Mazauric, CR, Inria Sophia Antipolis

cnrs

Inria informatics mathematics

PhD thesis
Inria Sophia Antipolis Méditerranée
Université Côte d'Azur, France

Graph problems motivated by (low and high) resolution models of large protein assemblies

Viet-Ha Nguyen

Directors:
Fréderic Havet, CNRS Research Director, I3S
Dorian Mazauric, Inria Researcher, Centre Inria d'Université Côte d'Azur

2021

Résumé

Pour comprendre les fonctions biologiques d'un assemblage moléculaire (AM), il est utile d'en avoir une représentation structurale. Celle-ci peut avoir deux niveaux de résolution : basse résolution (*i.e.* interactions moléculaires) et haute résolution (*i.e.* position relative et orientation de chaque sous-unité, appelée *conformation*). Cette thèse s'intéresse à trouver de telles représentations à l'aide de graphes.

Dans la première partie, nous cherchons des représentations basse résolution. Etant donné la composition des complexes d'un AM, notre but est de déterminer les interactions entre ses différentes sous-unités. Nous modélisons l'AM à l'aide d'un graphe : les sous-unités sont les sommets, les interactions entre elles sont les arêtes et un complexe est un sous-graphe induit. Utilisant le fait qu'une sous-unité n'a qu'un nombre limité d'interactions, nous arrivons au problème suivant. Pour un graphe F et un entier k fixés, étant donné un hypergraphe H et un entier s, MAX $(\Delta \leq k)$-F-OVERLAY consiste à décider s'il existe un graphe de degré au plus k tel qu'au moins s hyperarêtes de H induisent un sous-graphe contenant F (en tant que sous-graphe). La restriction au cas $s = |E(H)|$ est appelée $(\Delta \leq k)$-F-OVERLAY. Nous donnons une dichotomie de complexité (P vs. NP-complet) pour MAX $(\Delta \leq k)$-F-OVERLAY et $(\Delta \leq k)$-F-OVERLAY en fonction du couple (F, k).

Dans la seconde partie, nous nous attaquons à la haute résolution. Nous sont donnés un graphe représentant les interactions entre sous-unités, un ensemble de conformations possibles pour chaque sous-unité et une fonction de poids représentant la qualité de contact entre les conformations de deux sous-unités interagissant dans l'assemblage. Le problème Discrete Optimization of Multiple INteracting Objects (DOMINO) consiste alors à trouver les conformations pour les sous-unités qui maximise une fonction d'utilité globale. Nous proposons une nouvelle approche à ce problème en relâchant la fonction de poids, ce qui mène au problème de graphe CONFLICT COLORING. Nous donnons tout d'abord des résultats de complexité et des algorithmes (d'approximation et à paramètre fixé). Nous menons ensuite des expérimentations sur des instances de CONFLICT COLORING associées à des diagrammes de Voronoi dans le plan. Les statistiques obtenues nous informent sur comment les parmètres de notre montage expérimental influe sur l'existence d'une solution.

Mots clés: résolution d'assemblage de protéines, biologie structurale, complexité, algorithme, graphe, hypergraphe, coloration.

Abstract

To explain the biological function of a molecular assembly (MA), one has to know its structural description. It may be ascribed to two levels of resolution: low resolution (*i.e.* molecular interactions) and high resolution (*i.e.* relative position and orientation of each molecular subunit, called *conformation*). Our thesis aims to address the two problems from graph aspects.

The first part focuses on low resolution problem. Assume that the composition (complexes) of a MA is known, we want to determine all interactions of subunits in the MA which satisfies some property. It can be modeled as a graph problem by representing a subunit as a vertex, then a subunit-interaction is an edge, and a complex is an induced subgraph. In our work, we use the fact that a subunit has a bounded number of interactions. It leads to *overlaying graph with bounded maximum degree*. For a graph family $\mathcal{F}$ and a fixed integer k, given a hypergraph $H = (V(H), E(H))$ (whose edges are subsets of vertices) and an integer s, MAX $(\Delta \leq k)$-F-OVERLAY consists in deciding whether there exists a graph with degree at most k such that there are at least s hyperedges in which the subgraph induced by each hyperedge (complex) contains an element of $\mathcal{F}$. When $s = |E(H)|$, it is called $(\Delta \leq k)$-F-OVERLAY. We present complexity dichotomy results (P vs. NP-complete) for MAX $(\Delta \leq k)$-F-OVERLAY and $(\Delta \leq k)$-F-OVERLAY depending on pairs (F, k).

The second part presents our works motivated by high resolution problem. Assume that we are given a graph representing the interactions of subunits, a finite set of conformations for each subunit and a weight function assessing the quality of the contact between two subunits positioned in the assembly. Discrete Optimization of Multiple INteracting Objects (DOMINO) aims to find conformations for the subunits maximizing a global utility function. We propose a new approach based on this problem in which the weight function is relaxed, CONFLICT COLORING. We present studies from both theoretical and experimental points of view. Regarding the theory, we provide a complexity dichotomy result and also algorithmic methods (approximation and fixed paramater tracktability). Regarding the experiments, we build instances of CONFLICT COLORING associated with Voronoi diagrams in the plane. The obtained statistics provide information on the dependencies of the existences of a solution, to parameters used in our experimental setup.

Keywords: resolution of protein assembly, structural biology, complexity, algorithm, graph, hypergraph, coloring.

Contents

Chapter 1

Introduction

1.1 Structural biology context to Graph Model

Structural modeling of macromolecular assemblies. All biological phenomena (cognition, immune response, metabolism, etc) are based on structural description of macromolecular assemblies (MA): complexes of interacting molecules (subunit), namely proteins or nucleic acids [Fer99]. As conveyed by the simplistic *lock-and-key* model, the association of two or more molecules forming a complex requires these partners to be complementary. In fact, structural studies of molecules are carried out under the *structure-dynamics-function paradigm*, as it is the structure and the dynamics of molecules which explain biological functions. However, while genome sequencing projects have produced hundreds of millions of genes, there are only of the order of 10^5 structures in the Protein Data Bank (`https://www.rcsb.org/`). Thus, little is known in terms of macro-molecular interactions, at the structural level. Structures of molecules and complexes may be ascribed to two levels of resolution. At low resolution, the only information available may consists of the pairwise interactions of the subunits in the MA. A higher resolution, the relative position and orientation of the individual subunits may be known. Such models culminate with atomic resolution models, where individual atomic positions are known precisely. Because each atom has three Cartesian coordinates, such systems involve hundreds of thousands of degrees of freedom. This high dimensionality is naturally very challenging from the modeling standpoint. Structure determination is equally challenging from the experimental / structural biology standpoint. There are a number of experimental techniques which help providing structural information both at low and high resolution. We now discuss them in the context of low and high resolution models.

Low resolution problem. Native mass spectrometry is an experimental technique well suited to gather low resolution information. Assume that the composition of a MA, in terms of individual subunits, is known. A given assembly can be chemically split into complexes by manipulating chemical conditions, and measuring the mass of these complexes makes it possible to infer which subunits participate to each complex. In this context, the CONNECTIVITY INFERENCE (CI) problem [ACCC15] aims at inferring plausible pairwise contacts between subunits in the MA, given the composition (in terms of subunits) of the individual complexes. Additional biological constraints can also be incorporated [CHM+19, HMNW20]. This problem can be conveniently modeled by graphs and hypergraphs. We consider the hypergraph H whose vertices represent the subunits and whose hyperedges are the complexes. We are then looking for a graph G with the same vertex set as H whose edges represent the contacts between subunits, and satisfying some properties based on biological structure constraints. Solving CONNECTIVITY INFERENCE yields an *interaction graph* $G = (V, E)$ that represents the macromolecular assembly. The vertices are the subunits (e.g. proteins) and the edges are the (plausible) connections between subunits. However, such a graph falls short from providing high-resolution information.

High resolution problem. The main techniques to gather high resolution information are X ray crystallography, nuclear magnetic resonance (NMR), an elite club recently joined by cryo-electron microscopy. These techniques usually deliver static snapshots of molecules and complexes. Therefore, understanding dynamic processes requires building and animate models from these static complexes. A convenient way to do so consists of synergizing two types of models, namely a low resolution model for a MA, and high resolution (atomic) models for its subunits. Assume that the low resolution model provides the interaction graph of its subunits. Also assume that a finite set of high resolution conformations has been computed for each of the individual subunits. A route of choice to build a high resolution model of the MA consists of choosing and assembling individual conformations for each subunit. In doing so, we assume that a *scoring function* assessing the quality of the contact between two subunits positioned in the assembly is available. This problem was formalized using Discrete Optimization of Multiple INteracting Objects (DOMINO) in [LTSW09], so as to pick positions for the subunits maximizing a global utility function.

Indeed, the higher the score, the better connection of an assembly in global. A way to see this as a graph problem is using coloring where a set of conformations for each subunit is represented as a color set of the corresponding vertex in the interaction graph. We are given a graph, each vertex v receives a set of possible colors $C(v)$, and a function w that assigns a weight $w((u,i),(v,j))$ in $\mathbb{R} \cup \{-\infty\}$, for each edge $uv \in E(G)$, and every pair of colors $i \in C(v), j \in C(u)$. Hence, for each coloring of the graph, every edge is assigned a weight, and the DOMINO problem consists in deciding whether there exists a coloring of G which satisfies some properties.

1.2 Outline of this thesis

In this thesis, there are two parts corresponding to the previously mentioned two problems.

In Part I, we investigate the low resolution problem. Let G and H be respectively a graph and a hypergraph defined on a same set of vertices, and let F be a fixed graph. We say that G *F-overlays* a hyperedge S of H if F is a *spanning subgraph* of the subgraph of G induced by S (*i.e.* $|S| = |F|$ and $G[S]$ has a copy of F as a subgraph), and that G *F-overlays* H if it F-overlays every hyperedge of H. We exploit the fact that the number of contacts of a subunit is bounded, then the interaction graph has a bounded maximum degree. We study the computational complexity the following problem. Given a fixed graph F and a fixed integer k, MAX $(\Delta \leq k)$-F-OVERLAY, is given a hypergraph H and an integer s, consists in deciding whether there is a graph with maximum degree at most k that F-overlays at least s hyperedges of H. And a particular case when $s = |E(H)|$, $(\Delta \leq k)$-F-OVERLAY consists in deciding whether there is a graph with maximum degree at most k that F-overlays H. These problems will be precisely presented in Chapter 2. We give a complete polynomial/NP-complete dichotomy for the MAX $(\Delta \leq k)$-F-OVERLAY problems depending on the pairs (F, k). Then we study $(\Delta \leq k)$-F-OVERLAY in the next chapter which includes polynomial proofs of several pairs (F, k) in Section 4.1 but mainly in the next two sections, we aim to prove that for any graph F which is neither complete nor anticomplete, there exists a positive integer k_0 such that $(\Delta \leq k)$-F-OVERLAY is NP-complete for all $k \geq k_0$. Last, we investigate $(\Delta \leq k)$-F-OVERLAY restricted on the class of hypergraphs called *neat hypergraphs* (in which any two hyperedges intersect in at most one vertex).

In Part II, the high resolution problem is studied by first introducing formally DOMINO and then giving a new approach based this problem in which the weight function is relaxed. We denote by $[k]$ the set of integers $\{1, \dots, k\}$. In DOMINO, given a graph G, each edge $uv \in E(G)$ is given an edge-weighted complete bipartite graph K^k_{uv} (which is the complete bipartite graph $K_{k,k}$ on vertex sets $([k], [k])$ with an edge-weighted function). We denote by (u, i) the color i at vertex v in this graph, for $i \in [k]$. In the latter, the edge-weighted function is reduced by fixing a weight threshold λ. For any edge $uv \in E(G)$ with K^k_{uv}, an edge $(u, i)(v, j)$ is then *fulfilled* if its weight is at least λ and *conflict* otherwise. And so the graph which contains all the conflict edges is called the *conflict graph* of the edge uv. Given a graph G and two integers k, q, and the set of conflict graphs the FULFILL COLORING problem consists in determining whether there is a coloring of G such that the number of fulfilled edges is at least q, and CONFLICT COLORING consists in deciding whether there exists a coloring of G with non-conflict edges. The particular case of these problems when k is fixed are called FULFILL k-COLORING, CONFLICT k-COLORING. We then establish the complexity dichotomies (P vs NP-complete) for FULFILL k-COLORING, CONFLICT k-COLORING and some of their restrictions in Chapter 6. The next chapter includes some algorithms which we give an approximation one to FULFILL 2-COLORING and show a polynomial-time algorithm that solves DOMINO for graphs with bounded treewidth. Finally, in Chapter 8, we present experimental works in which we build instances of CONFLICT COLORING associated with *Voronoi diagrams* in the plane. The obtained statistics provide information on the dependencies of the existences of a solution, to parameters used in our experimental setup.

1.3 Preliminaries

1.3.1 Definitions

Graphs

A *graph* G is an order pair (V, E) where V is the finite set of elements called *vertices* and E is the set of *edges* in which each is a pair of vertices, which are also called *endvertices* (or *endpoints*). The vertex set and edge set are respectively referred to as $V(G)$ and $E(G)$. We denote by $|G|$ the *order* of G which is the size of the vertex set $|V(G)|$. For an edge uv, we say u, v are *adjacent* to each other and are *neighbors*. The *neighborhood* of a vertex v, denoted by $N_G(v)$, or simply $N(v)$ when G is clear from the context, is the set of vertices adjacent to v and its *degree*, denoted by $d_G(v)$ or simply $d(v)$, is the cardinality of $N_G(v)$. If G is such that all its vertices have the same degree r, then G is an r*-regular graph*. A vertex is *isolated* in G if it has degree 0. The minimum and maximum degree of G are respectively denoted by $\delta(G)$ and $\Delta(G)$. Hence a graph F has no isolated vertices if and only if $\delta(F) \geq 1$. We denote by V_i (resp. $V_{\leq i}$, $V_{\geq i}$) the set of vertices of G that has degree exactly (resp. at most, at least) i in G. A graph H is a *subgraph* of G if $V(H) \subseteq$ and $E(H) \subseteq E(G)$. If $E(H) = \{uv \in E(G) \mid u, v \in V(H)\}$, then H is an *induced subgraph* of G. A *connected component* is an induced subgraph in which there is a path between any two vertices.

An *independent set* of graph is a set of pairwise non-adjacent vertices. A graph is *complete* (resp. *anticomplete*) if its vertices are pairwise adjacent (resp. non-adjacent). The complete (resp. anticomplete) graph on p vertices is denoted by K_p (resp. $\overline{K}_p$). Since the latter has no edges, it is also called an *edgeless graph*. We denote by P_t the path on t vertices. For $v \in V(G)$, $G - v$ is the graph obtained by removing v and its incident edges. The disjoint union of two graphs G_1, G_2 is denoted by $G_1 + G_2$. A *matching* of a graph is a set of independent edges (*i.e.* pairwise non-intersecting edges). A matching is *maximum* if it contains the largest possible of edges. A graph G is *bipartite* if its vertex set $V = A \cup B$ a partition into two disjoint sets A, B and any edge has an endvertex in A and the other in B, often write $G = ((A, B), E)$

A directed graph consists of a vertex set and a set of directed edges (arcs). A *strongly connected component* (for directed graph) is a subgraph in which there is a directed path from every vertex to every other.

Hypergraphs

Let a hypergraph H. We denote by $V(H)$ and $E(H)$ respectively its sets of vertices and hyperedges where a hyperedge is a subset of vertices. A hypergraph is *p-uniform* for some $p \in \mathbb{N}$, if all its hyperedge have exactly p vertices. A hypergraph is *neat* if any two distinct hyperedges intersect in at most one vertex. We denote by $K(H)$, the graph obtained by replacing each hyperedge by a complete graph. In other words, $V(K(H)) = V(H)$ and $E(K(H)) = \{xy \mid \exists S \in E(H), \{x,y\} \subseteq S\}$. The *edge-weight function* induced by H on $K(H)$, denoted by w_H, is defined by $w_H(e) = |\{S \in E(H) \mid e \subseteq S\}|$. In words, $w_H(e)$ is the number of hyperedges of H containing e. A hypergraph H is *connected* if $K(H)$ is connected, and the connected components of a hypergraph H are the connected components of $K(H)$.

Complexity classes P and NP

In computational complexity theory, P is one of fundamental class where any decision problem in this class can be solved in polynomial time. Many problems are known in P including the decision versions such as finding a maximum matching of a graph [Sch03], and deciding whether two given vertices of a directed graph are in the same strongly connected component [Tar72].

NP is the class of decision problems in which any "yes" instance has a polynomial size certificate and certificates can be checked in polynomial time. Thus P is included in NP. The hardest problems in NP are NP-complete, a notion first introduced in 1971 by Cook and Levin [Coo71]. A decision problem is NP-*hard* if any NP problem can be reduced to it in polynomial time, and NP-*complete* if it is also in NP. As a fundamental consequence, if we can solve efficiently (*i.e.* in polynomial time) any NP-complete problem, so do we for all NP problem.

A problem is proven to be in P by either directly giving a polynomial time algorithm or reducing to a P problem. In this case the reduction from problem A to problem B in P gives an upper bound on the complexity of problem A. However, NP-completeness gives the lower bound of complexity because showing the NP-completeness of a problem B, we build a polynomial time reduction from a NP-complete problem A to B which implies that B is at least as difficult as A. The first NP-complete problem, known as SAT (see section 1.3.3 below), was directly proven from the definition of NP using *Turing machines*.

In this thesis, we use complexity results of some typical problems for our complexity proofs. The problem INDEPENDENT SET (decision version) is given a graph and an integer α, decides that there exists an independent set with at least α vertices. It is proven by Karp in [Kar72], to be NP-complete even for cubic graphs (*i.e.* 3-regular graphs). And mostly, we use 3-SAT, (PROPER) COLORING for reductions in NP-complete proofs and MATCHING for polynomial results.

1.3.2 Matching problems

A matching in a graph is a set of pairwise non-intersecting edges. For the sake of simplicity, we say that a vertex belongs to a matching M when it is an endvertex of an edge in M .We also say that it is *matched* (*covered or saturated*). A vertex that is not matched by M is *unmatched*. A matching is *maximal* if it is not included in any other matching; and a matching is *maximum* if it contains the largest possible number of edges, so obviously maximal. A matching which covers all vertices of a graph is a *perfect matching*. A maximal matching can easily be found using a greedy algorithm: start from an empty set M, and as long as it is possible, add to M an edge in G which is *independent* (i.e. has no common endvertex) with all the edges already in M. But in many problems, we need to find a maximal matching that optimizes some properties. A first example is the problem MAXIMUM MATCHING which consists in finding a maximum matching that is an a maximal matching with the maximum number of edges. A generalization of MAXIMUM MATCHING to edge-weighted graphs (that graphs equipped with a weight function on the edges) is MAXIMUM-WEIGHT MATCHING: given an edge-weighted graph, find a matching with maximum weight (i.e. sum of weights of its edges).

MAXIMUM MATCHING. This problem is a very classic optimization problem in graph theory which finds a maximum matching of given a graph. It can be solved in polynomial time. A key notion to do so, is the one of "augmenting path". Let M be a matching in a graph G. An *M-alternating path* is a path along which the edges are alternately in M and not in M. An *M-augmenting path* is an alternating path whose endvertices are unmatched. Let P be an M-augmenting path, a new matching M' is obtained by replacing edges in $P \cap M$ by all edges in $E(P) \setminus (E(P) \cap M)$. Thus $M' = (M \setminus (E(P) \cap M) \cup (E(P) \setminus M)) = M \triangle E(P)$ and we have $|M'| = |M| + 1$ (Figure 1.1). Hence, if a matching is maximum, then there is no augmenting paths. By **Berge's lemma**, this necessary condition is also sufficient.

Lemma 1 ([Ber57]). *M is maximum if and only if there is no M-augmenting paths.*

This is the main idea of many polynomial-time algorithms for finding a maximum matching. And we can describe an algorithm as follows.

1. Initialize a matching $M := \emptyset$.
2. While *there exists an M-augmenting path P* do $M := M \triangle E(P)$.

Figure 1.1: **Example of an M-augmenting path.** In the figure to the left, there is a matching M of two blue edges, the two red vertices are unmatched, and there is an M-augmenting path P (dashed red) between these two unmatched vertices. We obtain a new matching with one more edge by removing blue edges in P from M and adding the other black edges in P to M. We then get the matching depicted in blue on the figure to the right.

In bipartite graphs. MATCHING in bipartite graphs is an important special case by its practical roles in real world [Rob03] and even is the first studies of MATCHING problems in general, known as **Hall's (marriage) theorem**. Imagine that there are two groups of men and women. Each woman has a set of men, any of which and the woman are happy to marry. The marriage theorem answers whether it is possible to match the woman and man so that everyone is happy.

Let a bipartite graph $G = ((A, B), E)$ with $|A| < |B|$. If there is a matching covering all vertices of A then it is maximum. Hence, we want to answer whether there exists a matching which covers A. It is necessary that for any $S \subseteq A$, $|S| \leq |N(S)|$ where $N(S)$ is the set of vertices that are adjacent to at least one vertex of S. The fact that this is also sufficient is known as **Hall's theorem**.

Theorem 2 ([Hal35])**.** *Let $G = ((A, B), E)$ be a bipartite graph. G has a matching covering all vertices of A if and only if $|N(S)| \geq |S|$ for all $S \subseteq A$.*

When $|A| = |B|$, his result applies to perfect matching. In bipartite graphs, in order to find a maximum matching in the view of Lemma 1, we have to find augmenting paths. There is a simple way to solve it as follows: given $G = ((A, B), E)$ and M an arbitrary matching, we build a directed graph D_M by orienting any edge uv (for $u \in A$, $u \in B$) from v to u if $uv \in M$, otherwise from u to v. Now, finding a directed path in D_M from a vertex in $A \setminus M$ to a vertex in $B \setminus M$ if any is equivalent to finding an M-augmenting path. Since such a path can be found in time $O(|E|)$ (*e.g.* by using breadth first search) and there are at most $|A|$ such paths. A maximum matching in a bipartite graph can be found in time $O(|E||V|)$.
There are many faster algorithms than this one to solve MAXIMUM MATCHING. One of them, due to Hopcroft and Karp in 1973 [HK73] runs in $O(|E|\sqrt{|V|})$. This improvement searches for a maximal set of augmenting paths, and makes only $O(\sqrt{|V|})$ such searches.

In general graphs. The idea of using augmenting path is still a key for finding a maximum matching in general graphs. As described in the algorithm above, in bipartite graphs, we can always orient the edges, whereas we cannot in general graphs (because of odd cycles). To solve this problem, Edmonds introduced in 1965 the term *blossom*. An M-alternating path $P = v_0 v_1 \dots v_t$ where t is odd, $v_0, \dots v_{t-1}$ are distinct, v_0 is unmatched by M and $v_t = v_i$ for some even $i < t$. The cycle $v_i v_{i+1} \dots v_t$ is an M-blossom. Observe that if there is an augmenting path $v_0 v_1 \dots v_i \dots v_j v_{j+1} \dots v_{t'}$ where $v_i \dots v_j$ is a subpath of P, and we identify all vertices in the blossom to a new vertex v_B (so contract the blossom), then we have $v_0 \dots v_B v_{j+1} \dots v_{t'}$ which is also an augmenting path (see Figure 1.2 for an example). It is the core of Edmonds' algorithm. Define G' the contracted graph obtained from G by contracting an M-blossom, and M' is the matching $M \cap G'$. We have the following.

Theorem 3. *G has an M-augmenting path if and only if G' has an M'-augmenting path.*

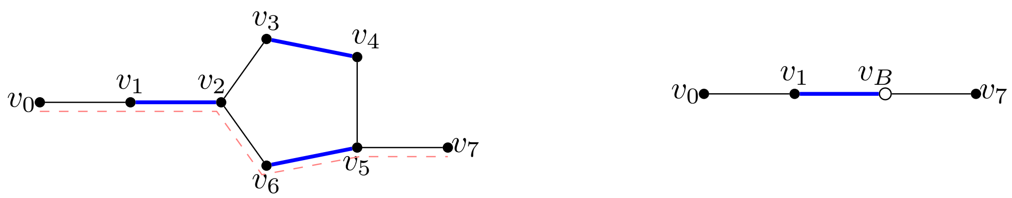


Figure 1.2: **Example of blossom.** The cycle $B = v_2 \dots v_6 v_2$ is a blossom with respect to a matching in blue and there is an augmenting path P (dashed red) (left figure). The right figure shows the contraction of the blossom to a new vertex v_B. This changes P to P' which is still an augmenting path.

The following problems are two of generalizations of MAXIMUM MATCHING.

MAXIMUM k-MATCHING Given a graph G, a *k-matching* M is a subgraph of G with maximum degree at most k. Thus, a matching is the case when $k = 1$. MAXIMUM k-MATCHING is the problem of finding a k-matching with the maximum number of edges. There are several poly-time algorithms for finding a maximum k-matching. An easy one is to reduce it to MATCHING. Given a graph G, we build a graph G' as follows. For each vertex $v \in V(G)$, we add k vertices $v_1, \dots, v_k$ to $V(G')$; and for each edge $uv \in E(G)$, we add k^2 edges connecting the k vertices corresponding to u to those corresponding to v.

Note that this construction also work to find a subgraph of a bipartite graph $G = ((A, B), E)$ such that all vertices in A have degree at most a and those in B have degree at most b.

MAXIMUM-WEIGHT MATCHING Let G be a graph with edge-weighted function, the problem finds a matching with maximum weight. MATCHING is the case when the weight of every edge is 1. In MAXIMUM-WEIGHT MATCHING, the notation of augmenting path is the same which is a path alternating edges in and not in a matching and its two endpoints are unmatched. In bipartite graph, it is ASSIGNMENT PROBLEM and typically **Hugarian's method** is one of algorithms solving this problem. This algorithm finds augmenting paths by using a modified shortest path search. Given a bipartite graph $G = ((A, B), E)$ with an edge-weighted function w and a matching M. Let D_M be the directed graph obtained from G by orienting each edge $uv \in M$ (for $u \in A, v \in B$) from v to u with length $l_{\overrightarrow{vu}} := w_{uv}$, otherwise from u to v with length $l_{\overrightarrow{uv}} := -w_{uv}$. Hence, we aim to find a shortest path P from $A \setminus M$ to $B \setminus M$ (so P is an augmenting path) with negative length (which then increases the total weight of new matching after exchanging edges in P).

In non-bipartite graphs, the problem is much more complicated. And it can be solve by using the unweighted Edmonds' algorithm as a subroutine.

1.3.3 Boolean Satisfiability

Boolean Satisfiability problem (SAT) determines, given a boolean formula, if there exists an assignment which satisfies the formula. It is known as the first NP-complete problem, proved by Cook-Levin in 1970s [Coo71]. An instance of SAT is combined from variables and operators (disjunction $\vee$, conjunction $\wedge$, negation $\neg$) with parentheses. Note that, we use both $\neg x$, $\bar{x}$ as the negation of a variable x).
SAT for disjuntive normal form is obvious, indeed such a formula is satisfiable if and only if at least one clause is satisfiable which is equivalent to that no variable and its negation belong to the same clause.
Every SAT formula can be transformed to an equivalent conjunctive normal form (CNF) (which may be exponentially longer). The complexity results are related to SAT for CNF formulas, which is also called CNF-SAT. Hence, when we say a SAT instance, we mean a CNF-SAT one.

A restriction of the problem is on the number of literal per clause. k-SAT is when the number of literal per clause is exactly k. As shown by Cook [Coo71], 3-SAT is NP-complete which is the tightest possible on the number of variables per clause because 2-SAT is in P.

Theorem 4 ([ES76]). *2-SAT is in P.*

Proof sketch. We use an *implication graph* which is a directed graph. There are two vertices per variable (one for this variable x_i and one for its negation $\neg x_i$). Each clause gives pair of implications on its two variables (*e.g.* $(x_1 \vee \neg x_2)$ gives arcs $\neg x_1 \rightarrow \neg x_2$ and $x_2 \rightarrow \neg x_1$). We can build such an implication graph from any instance of 2-SAT, in polynomial time. An instance is satisfiable if and only if no variable and its negated are in the same strongly connected component of the implication graph. This can be checked in linear time [APT79]. □

According to the algorithm used in [APT79], if a 2-SAT formula is satisfiable, then we can also find in linear time an assignment satisfying it. The implication graph of a formula gives chains of implications which help to order variables and make choice to them.

3-SAT can be seen as the starting point leading to a number of problems which are NP-complete. Those first ones are Karp's list of NP-complete problems [Kar72] and then many of studied problems are variants of 3-SAT. It is pointed out that studies on the dichotomy of (strongest possible) restrictions of a problem help understanding the boundary between P and NP-complete cases of a problem, as well as establishing the NP-completeness of problems which may be easier to reduce from, for future studies [HS78]. (r,k)-SAT is the class of instances of k-SAT where every variable appears in at most r clauses. It is proven in [Tov84] that (3,4)-SAT is NP-complete.

Theorem 5 ([Tov84])**.** *(3,4)-SAT is NP-complete.*

A usual way to transform a SAT formula to another formula with restrictions on the number of literals per clause or the number of occurrences of a variable, is to add new variables in order to validate these conditions, and then add new clauses enforcing constraints on these new variables.

Proof sketch. A reduction from 3-SAT. Let Φ be a formula of this problem with n variables and m clauses. We build a formula Φ' of (3,4)-SAT as follows. For each variable x which appears in $r \in [4]$ clauses, create r new vertices $x_1, \dots, x_r$ and replace the i-th occurrence of x with x_i in Φ'. There are m clauses.
We want that x_i for $i \in [r]$ are all true or all false, so that they correspond to one original variable. To enforce this, we use the following structure: $\bigwedge_{i\in[r]}(x_i \vee \bar{x}_{i+1})$ with subscripts taken modulo r. Since each of these clauses $(x_i \vee \bar{x}_{i+1})$ has only two literals, we introduce a new literal y_i, and add the clause $(x_i \vee \bar{x}_{i+1} \vee \bar{y}_i)$ to Φ'. There are $3m$ clauses.
We now enforce y_i to be true by adding the following clauses to Φ':

- $(y_i \vee a_i \vee b_j)$ for $j \in [3]$,
- $(d_j \vee a_j \vee b_j)$, $(d_j \vee a_j \vee \bar{b}_j)$, $(d_j \vee \bar{a}_j \vee b_j)$ for $j \in [3]$,
- $(\bar{d}_1 \vee \bar{d}_2 \vee \bar{d}_3)$.

There are 13 clauses for each y_i, and there are $3m$ such variables. Thus, Φ' contains $43m$ clauses in total. And Φ' is satisfiable if and only if so is Φ. □

If repetitions of a variable in a clause is not allowed, then Theorem 5 is optimal, because (r,r)-SAT is solvable in constant time (Theorem 6 below). Note however that if clauses are allowed to have size 2 or 3, then the satisfaction problem with at most 3 occurrences per variable is NP-complete [Tov84].

Theorem 6 ([Tov84])**.** *(r,r)-SAT is satisfiable for any integer r.*

Proof sketch. Let Φ be a formula with the set X of n variables and the set C of m clauses. We construct the bipartite graph G such that the vertex set is $X \cup C$ and there is an edge between variable $x_i \in X$ and clause $c_j \in C$ if and only if a literal of x_i belongs to c_j in Φ. Now (r,r)-SAT is satisfiable if and only if there exists a C-perfect matching of G, *i.e.* every vertex of C is in the matching. It turns out that such a matching exists, by Hall's marriage theorem: for any $C' \subseteq C$ we have $|C'| \leq |N(C')|$, otherwise a vertex in X has degree strictly bigger than r, *i.e.* a variable has strictly more than r occurrences. $\square$

Many other variants of SAT were studied which relate to the constraints on solutions, instances or operators such as 1-IN-3-SAT (exactly one true literal per clause), NAE-3-SAT (no clause has all three true literals), HORN-SAT (each clause has at most one positive literal), XOR-SAT (each clause contains XOR rather than OR operators). For these mentioned problems, the two first are NP-complete and the latter three are in P. The restrictions above are particular cases of **Schaefer's dichotomy theorem** which states necessary and sufficient conditions, such that a finite set of relations over Boolean domain generates a SAT variant which is NP-complete or solvable in polynomial time. Schaefer's dichotomy theorem allows us to determine the complexity results of many variants of SAT problems, but it does not apply to all variants: (r,k)-SAT is a typical example because given a set of relations, there is no way to restrict the occurrences of a variable.

Remark that not all problems may be either in P or NP-hard. Ladner's theorem states that, under the assumption P $\neq$ NP, there exits an infinite hierarchy of classes between P and NP (called NP-intermediate, problems in NP but neither in P nor NP-hard).

1.3.4 Graph coloring

A coloring of G is a function from $V(G)$ to a finite set of labels, called *colors*. If the number of used colors is k, then we call it a *k-coloring*. A coloring is *proper* if any two adjacent vertices in G have different colors. If G admits a proper k-coloring, then G is *k-colorable*. If each vertex of G is assigned a different color, then $k = |V(G)|$ and this coloring is trivially proper. An interesting question is to minimize k such that G is k-colorable, and the smallest such k is called the *chromatic number* of G, denoted $\chi(G)$. The decision version of this problem, called COLORING, consists in, given a graph G and an integer k, deciding whether G k-colorable? It is a typical one of NP-hard problems shown by Karp [Kar72]. However, this problem is obvious for some special classes of graphs. For example, the complete graph K_p has $\chi(K_p) = p-1$, or $\chi(\overline{K}_p) = 1$ for anticomplete graphs $\overline{K}_p$, and the chromatic number of bipartite graphs is 2. Bipartite graphs are by definition the 2-colorable graphs. They can be recognized in polynomial time and that a 2-coloring of a bipartite graph can be found in polynomial time. In contrast, it is NP- complete to recognize the 3-colorable graphs and given an 3-colorable graph, it is NP-hard to find a 3-coloring and even a 4-coloring.

An alternative way of deciding the chromatic number is to study the decision version of this problem, called *k-coloring problem* which decides whether a graph G is k-colorable. It is obvious when $k = 1$, the family of anticomplete graphs is the only 1-colorable. And for $k = 2$, a graph is 2-colorable if and only if it is bipartite which can be checked in polynomial-time. Alternatively, we can prove 2-coloring problem is in P by reducing to another P problem.

Theorem 7. 2-COLORING *is in* P.

Proof. We will reduce to 2-SAT. Let G be a graph with $V(G)$ of n vertices and $E(G)$ of m edges. We construct a formula Φ on n variables and $2m$ clauses as follows. For each vertex $v \in V(G)$, there is a variable x_v. And for each edge $uv \in E(G)$, we add two clauses $(x_u \vee x_v), (\bar{x}_u \vee \bar{x}_v)$ to Φ. These two clauses enforce that x_u, x_v must be different which corresponds to that u, v have different colors. Hence, G is 2-colorable if and only if Φ is satisfiable. In this case, let ϕ be a truth assignment of Φ, a 2-coloring c of G can be defined by taking $c(v) = 1$ (resp. 2) if $\phi(x_v) = true$ (resp. $false$). □

Hence a 2-coloring can be found through a satisfying assignment of a 2-SAT formula. Alternatively, we can also find a 2-coloring by partitioning the vertex set into two subsets (bipartite graph).

However, only these two cases which are polynomial-time solvable.

Theorem 8 ([GJS76])**.** 3-COLORING *is* NP*-complete.*

Proof. A reduction from 3-SAT. Given a formula Φ of this problem with n variable x_i for $i \in [n]$ and m clauses, we construct a graph G as follows. The vertex set $V(G)$ contains 3 vertices t, f, b; two vertices $v_i, \bar{v}_i$ for each variable x_i; and six vertices $y_j, y_j^1, \ldots, y_j^5$ for each clause C_j. G has two types of gadgets as described in Figure 1.3: one *variable gadget* and a *clause gadget* for each clause. And, we add edges y_jb, y_jf for $j \in [m]$.

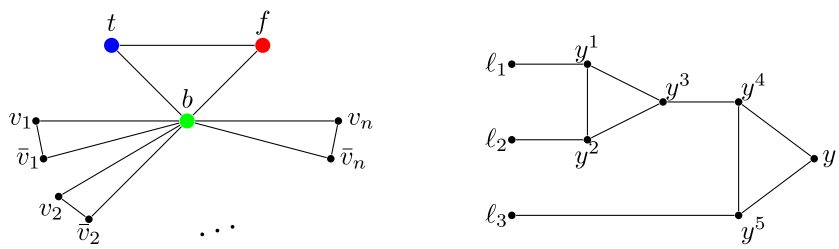


Figure 1.3: **Construction of graph G.** Variable gadget (left) and a clause gadget for $(\ell_1 \vee \ell_2 \vee \ell_3)$ (right) where each literal ℓ_i is corresponding to a *literal vertex* ℓ_i for $i \in [3]$ and the output of the clause is represented by the *output vertex* y.

First, G can be build in polynomial time. Observe that t, f, b must have 3 different colors 1, 2, 3 respectively, hence every y_j has color 1 since it is adjacent to b and f. By the construct of clause gadgets, any 3-coloring of this gadget satisfies that y has color 1 if and only if there exists $i \in [3]$ such that l_i has color 1.
Now, let c be a coloring of G, we define an assignment ϕ of Φ by taking $\phi(x_i) = true$ (resp. $false$) if $c(v_i) = 1$ (resp. 2). And so ϕ satisfies Φ.
Let ϕ be an assignment satisfying Φ, a coloring c of G is defined by taking $c(v_i) = 1$ (resp. 2) if $\phi(x_i) = true$ (resp. $false$). Since there is at least one true literal in each clause, then the corresponding vertex has color 1 which implies that the output vertex is colored 1 by c. Thus, c is a 3-coloring of G. □

In 1941, Brooks gave a relationship between the maximum degree of a graph and its chromatic number.

Theorem 9 ([Bro41]). *Let G be a connected graph, then $\chi(G) \leq \Delta(G)$, except for G an odd cycle or a complete graph, $\chi(G) = \Delta(G) + 1$.*

According to this theorem, every connected cubic graph except K_4 is 3-colorable. However, it becomes hard for graphs of maximum degree 4.

Theorem 10 ([Hol81]). 3-COLORING *is* NP*-complete on 4-regular graphs.*

We then obtain that k-COLORING is NP-complete for any $k \geq 4$ by an inductive proof whose base case is 3-COLORING. From a graph G of $(k-1)$-COLORING, we can construct a graph G' of k-COLORING by adding a universal vertex (this new vertex connects to every vertex of G). Then, it must be colored by a new color k. Therefore, $\chi(G') = \chi(G) + 1$.

Corollary 11. k-COLORING *is* NP*-complete for* $k \geq 3$.

A graph is *planar* if it can be drawn in the plane without any edge crossing. Related to these graphs, **Euler's formula** is a very well-known equation which relates the number of vertices, edges and faces in planar graphs, respectively denoted by n, m, f:

$$n - m + f = 2.$$

Since a face is bound by at least 3 edges and any two adjacent faces share an edge, then $3f \geq 2m$. Using Euler's formula, we obtain $3n - 6 \geq m$ as a corollary. Moreover, gathering with the following fact that the total degree of vertices is twice the number of edges, $\sum_{v \in V(G)} d(v) = 2m$, as a consequence there is a vertex in G of degree at most 5. The **Five Color Theorem** was found down in 1890 by P.J. Heawood.

Theorem 12 ([Hea90]). *Every planar graph is 5-colorable.*

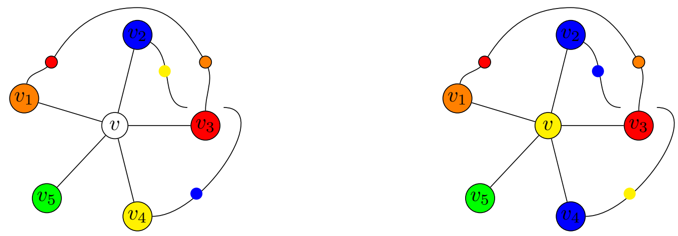


Figure 1.4: **Example of coloring vertex v.** v is not yet colored and 5 neighbors of v have different colors (left). Since there is no path between v_2 and v_4 in the induced subgraph by yellow and blue, then recolor v_4 (and yellow vertices in the subgraph) blue and color v yellow (right).

Proof. By induction on the number of vertices.
First, it is obvious to color a graph with at most 5 vertices using 5 colors.

Suppose that $n > 5$. Assume that v is a vertex of degree at most 5. By induction, we know that $G - v$ is 5-colorable.
If $d(v) \leq 4$, then we color v with a color different from its neighbors, which is a 5-coloring of G. If $d(v) = 5$, and if its neighbors are colored with at most 4 colors, then it is easy to color v. Assume now the neighbors of v, say $v_1, \ldots, v_5$ (in clockwise order) have colors 1,…,5, respectively. Let G_{ij} be the induced subgraph of G on vertices having colors i, j. If there is no path from v_1 to v_3 in G_{13}, then we can change the color of v_1 to 3, and switch colors of vertices in the same component as v_1 in G_{13}, so we can give color 1 to vertex v.
If v_1, v_3 are in the same connected component of G_{13}, then there is no path between v_2 to neither v_4 nor v_5 because of the planarity of G (Figure 1.4). Hence, we can apply the same reasoning as the previous case. □

While the Five Color Theorem is rather simple, it is not the case for the **Four Color Theorem**. It was first proven by Appel and Haken around a century later, in 1976.

Theorem 13 ([AH89, AH77a, AH77b])**.** *Every planar graph is 4-colorable.*

In 1996, [RSST96] presented a quadratic-time algorithm finding a 4-coloring on planar graphs, they also gave a linear-time algorithm which finds a 5-coloring on these graphs. One year later, they improved the proof of the Four Color Theorem shown by Appel and Haken, with a simpler and clearer proof. In spite of that, 3-coloring on planar graphs is NP-complete. Namely,

Theorem 14 ([Dai80])**.** 3-COLORING *is* NP*-complete on planar 4-regular graphs.*

There are many other studies of the coloring problem on particular families of graphs: dense graphs [Edw86], and recently graphs of H-free subgraphs [HKL$^+$10, BCM$^+$18, GJPS17].

Reasearchers have also generalized the coloring problem. For example, PRECOLORING EXTENSION where a coloring c is given on a subset of vertices of a graph G and decides whether c can be extended to a coloring of G [Bod93] (in coloring problem the precoloring c is empty), LIST COLORING where each vertex is given a list of admissible colors and the coloring must give to each vertex a color from its list [GJPS17] (in k-coloring problem the list of each vertex is $[k]$), or H-coloring which is a coloring with respect to a fixed graph H whose vertices represent colors and edges give the binary relation of allowed neighboring colors [HN90] (in k-coloring problem we have $H = K_k$). In Part II of this thesis, we investigate another generalization of coloring problem, CONFLICT COLORING.

Part I

Overlaying a hypergraph with a graph

Chapter 2

Introduction

2.1 Approaches of Connectivity Inference problem

In order to obtain the low resolution structure of molecule-macro assemblies, the following problem, known as CONNECTIVITY INFERENCE arose. The list of subunits is known, and given a set of complexes, one has to determine the plausible contacts between subunits of an assembly. This problem can be modeled using a hypergraph: each of its vertices represents a subunit and a hyperedge represents a complex. Then the aim is to find a graph G on the same vertex set whose edges associate to contacts between subunits and so must satisfy (i) some local properties for every complex (*i.e.* hyperedge), and (ii) some other global properties.

For example, consider the following properties: (i) the subset of vertices of every hyperedge must form a connected subgraph of G and (ii) there is an upper bound on the degree of any vertex. The first property comes from the fact that the complexes obtained (e.g. by native mass spectrometry) are connected components of a given assembly. The second property is deduced from a limited number of contacts of a subunit to the other in an assembly in practice. In our example in Figure 2.1, we use the first property and assume that the degree of every vertex does not exceed 3. Consider a hypergraph H whose hyperedges are $\{v_1, v_2, v_3\}$, $\{v_2, v_4, v_5\}$, $\{v_3, v_4, v_5\}$, and $\{v_4, v_5, v_6, v_7\}$ as depicted in Figure 2.1 (left) in which a vertex labeled i represents v_i. The graph G described in Figure 2.1 (right) satisfies the previous properties: the induced subgraph of G on every hyperedge is connected and the maximum degree of G is 3.

We first focus on the local properties. They are usually modeled by a (possibly infinite) family $\mathcal{F}$ of admissible graphs to which each complex must belong: to this

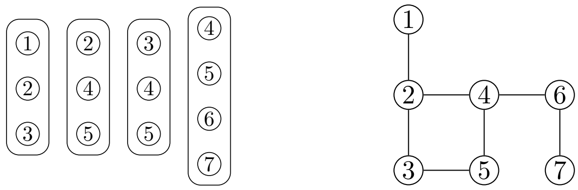


Figure 2.1: **Example of a hypergraph and an inference graph.** A hypergraph composed of seven vertices and four hyperedges (left). A graph G satisfying local properties (every hyperedge induces a connected subgraph in G) and a global property (every vertex has degree at most a given number which is 3 in this case).

end, we define the notion of *enforcement* of a hyperedge and a hypergraph. A graph G *$\mathcal{F}$-enforces* a hyperedge $S \in E(H)$ if the subgraph $G[S]$ of G induced by S belongs to $\mathcal{F}$, and G *$\mathcal{F}$-enforces* H if G $\mathcal{F}$-enforces all hyperedges of H. Very often, the considered family $\mathcal{F}$ is closed on taking edge supergraphs [ACCC15, CHM+19]: if $F \in \mathcal{F}$, then every graph obtained from G by adding edges is also in $\mathcal{F}$. Such a family is completely defined by its set $\mathcal{M} = \mathcal{M}(\mathcal{F})$ of minimal graphs that are the elements of $\mathcal{F}$ which are not edge supergraphs of any other. In this case, a graph G $\mathcal{F}$-enforcing S is such that there is an element of $\mathcal{M}$ which is a spanning subgraph of $G[S]$. This leads to the following notion of *overlayment* when considering minimal graph families.

Definition 15. A graph G *$\mathcal{F}$-overlays* a hyperedge S if there exists $F \in \mathcal{F}$ such that F is a spanning subgraph of $G[S]$, and G *$\mathcal{F}$-overlays* H if G $\mathcal{F}$-overlays every hyperedge of H.

As said previously, the graph sought will also have to satisfy some global constraints. Since in a macro-molecular assembly the number of contacts is small, the first natural idea is to look for a graph G with the minimum number of edges. This leads to the MIN-$\mathcal{F}$-OVERLAY problem: given a hypergraph H and an integer m, decide if there exists a graph G $\mathcal{F}$-overlaying H such that $|E(G)| \leq m$.

A typical example of a family $\mathcal{F}$ is the set of all connected graphs, in which case $\mathcal{M}(\mathcal{F})$ is the set of all trees. In [ACCC15], Agarwal et al. focused on MIN-$\mathcal{M}(\mathcal{F})$-OVERLAY for this particular family in the aforementioned context of structural biology. However, this problem was previously studied by several communities in other domains [CKN+15]. Indeed, it is also known as SUBSET INTERCONNECTION DESIGN, MINIMUM TOPIC-CONNECTED OVERLAY or INTERCONNECTION GRAPH PROBLEM, and was considered (among others) in the design of vacuum systems [DK95, DM88], scalable overlay networks [CMTV07, HHI+12, OR11], and reconfigurable interconnection networks [FHWE08, FW08]. Some variants have also been considered in the contexts of inferring a most likely social network [AAR10], determining winners of combinatorial auctions [CDS04], as well as drawing hypergraphs [BCPS10, JP87, KS03].

In [CHM+17, CHM+19], Cohen et al. presented a dichotomy regarding the polynomial vs. NP-hard status with respect to the considered family $\mathcal{F}$. Roughly speaking, they showed that the easy cases one can think of (e.g. when edgeless graphs of the right sizes are in $\mathcal{F}$, or if $\mathcal{F}$ contains only cliques) are the only families giving rise to a polynomial-time solvable problem: all others are NP-complete. They also considered the FPT/W[1]-hard dichotomy for several families $\mathcal{F}$.

2.2 Overlaying graph with bounded maximum degree

In this part of the thesis, we consider the variant in which the additional constraint is that G must have a bounded maximum degree: this constraint is motivated by the context of structural biology, since a subunit (e.g. a protein) cannot be connected to many other subunits. This yields the following problem for any family $\mathcal{F}$ of graphs and an integer k.

$(\Delta \leq k)$-$\mathcal{F}$-OVERLAY
Input: A hypergraph H.
Question: Does there exist a graph G $\mathcal{F}$-overlaying H such that $\Delta(G) \leq k$?

We denote a graph G $\mathcal{F}$-overlaying H with maximum degree at most k by $(\mathcal{F}, H, k)$-*graph* (resp. (F, H, k)-*graph*) for $\mathcal{F}$ is a graph family (resp. F a graph).

And we call $over_{\mathcal{F}}(H, G)$ the number of hyperedges of H that are $\mathcal{F}$-overlaid by G. A natural generalization is to find $over_{\mathcal{F}}(H, k)$, the maximum number of hyperedges $\mathcal{F}$-overlaid by a graph with maximum degree at most k.

MAX $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY
Input: A hypergraph H and a positive integer s.
Question: Does there exist a graph G such that $\Delta(G) \leq k$ and $over_{\mathcal{F}}(H, G) \geq s$?

Observe that there is an obvious reduction from $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY to MAX $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY (by setting $s = |E(H)|$).

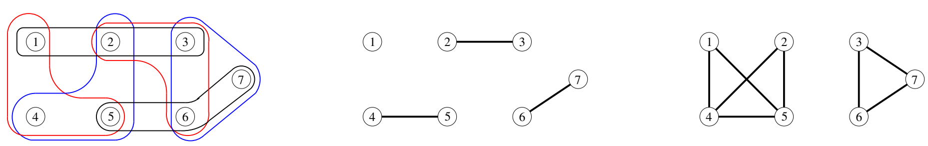


Figure 2.2: **Examples of $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY and MAX $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY.** In the figure, an instance H (left), a graph G with $\Delta(G) \leq 1$ that O_3-overlays H (with O_3 being the graph with three vertices and one edge) (center), and a $(C_3, H, 3)$-graph to MAX $(\Delta \leq 3)$-C_3-OVERLAY (with C_3 being the cycle on three vertices) (right).

2.3 Overview of this part

Our contribution is to provide the dichotomy of complexity (P vs NP) for MAX $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY and $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY considering pairs (F, k). We mainly consider the case when **the family $\mathcal{F}$ contains a unique graph** F. We abbreviate $(\Delta \leq k)$-$\{F\}$-OVERLAY and MAX $(\Delta \leq k)$-$\{F\}$-OVERLAY to $(\Delta \leq k)$-F-OVERLAY and MAX $(\Delta \leq k)$-F-OVERLAY, respectively. By definition those two problems really make sense only for $|F|$-uniform hypergraphs *i.e.* hypergraphs whose hyperedges are of size $|F|$. Therefore, we work often with hypergraphs that are uniform, without specifying it.

Remark 16. If F is a graph with maximum degree greater than k, then solving $(\Delta \leq k)$-F-OVERLAY or MAX $(\Delta \leq k)$-F-OVERLAY is trivial as the answer is always 'No'. So we only study the problems when $\Delta(F) \leq k$.

If F is an anticomplete graph (i.e. with no edges), then MAX $(\Delta \leq k)$-F-OVERLAY is also trivial, because for any hypergraph H, the anticomplete graph on $V(H)$ vertices F-overlays H. Hence the first natural interesting cases are the graphs with one edge which in Section 2.4, we present the complete dichotomy for this graph family.

Then, in Chapter 3, we give a complete polynomial/NP-complete dichotomy for the MAX $(\Delta \leq k)$-F-OVERLAY problems.

In Chapter 4, we investigate the complexity of $(\Delta \leq k)$-F-OVERLAY problem. We believe that each such problem is either polynomial-time solvable or NP-complete. However the dichotomy seems to be more complicated than the one for MAX $(\Delta \leq k)$-F-OVERLAY. So we first exhibit several pairs (F, k) such that $(\Delta \leq k)$-F-OVERLAY is polynomial-time solvable, while MAX $(\Delta \leq k)$-F-OVERLAY is NP-complete in Section 4.1. In the next two sections, we present the NP-complete results of $(\Delta \leq k)$-F-OVERLAY problem which we prove that except when F is complete or anticomplete, if k is large enough (with respect to F), then $(\Delta \leq k)$-F-OVERLAY is NP-complete.

Remark 17. Our aim is to prove that $(\Delta \leq k)$-F-OVERLAY is NP-complete under some assumptions on k and F. Observe that, given a graph G, we can easily check whether G F-overlays H or not in polynomial-time solvable, thus the problem is clearly in NP. Therefore, we only need to prove that the problem is NP-hard.

2.4 Warm up: graphs with one edge

In this section, we consider graphs with one edge which for each on $p \geq 2$ vertices, we denote by O_p, then we establish the following dichotomy theorem.

Theorem 18. *Let $k \geq 1$ and $p \geq 2$ be integers. If $p = 2$ or if $k = 1$ and $p = 3$, then* MAX $(\Delta \leq k)$-O_p-OVERLAY *and* $(\Delta \leq k)$-O_p-OVERLAY *are polynomial-time solvable. Otherwise, they are* NP*-complete.*

Let $p \geq 2$, and H be a p-uniform hypergraph. Consider the edge-weighted graph $(K(H), w_H)$. For every matching M with at least one edge of this graph, let $G_M = (V(H), M)$. Every hyperedge O_p-overlaid by G_M is a superset of at least one edge of M and at most $\lfloor \frac{p}{2} \rfloor$ edges of M. We thus have the following:

Observation 1. For every matching M of $K(H)$, we have:

$$\frac{1}{\lfloor \frac{p}{2} \rfloor} w_H(M) \leq over_{O_p}(H, G_M) \leq w_H(M), \tag{2.1}$$

where $w_H(M) = \sum_{e \in M} w_H(e)$.

Consider first the case when $p = 2$. Let H be a 2-uniform hypergraph. Every hyperedge is an edge, so $K(H) = H$. Moreover, a (hyper)edge of H is O_2-overlaid by G if and only if it is in $E(G)$. Hence MAX $(\Delta \leq k)$-O_2-OVERLAY is equivalent to finding a maximum k-matching (that is a subgraph with maximum degree at most k) in $K(H)$. This problem is polynomial-time solvable, see [Sch03, Chap. 31], hence:

Proposition 19. MAX $(\Delta \leq k)$-O_2-OVERLAY *is polynomial-time solvable for all positive integer k.*

If $p = 3$, Inequalities (2.1) are equivalent to $over_{O_3}(H, G_M) = w_H(M)$. Since the edge set of a graph with maximum degree 1 is a matching, MAX $(\Delta \leq 1)$-O_3-OVERLAY is equivalent to finding a maximum-weight matching in the edge-weighted graph $(K(H), w_H)$. This can be done in polynomial-time, see [Jun13, Chap. 14].

Proposition 20. MAX $(\Delta \leq 1)$-O_3-OVERLAY *is polynomial-time solvable.*

We shall now prove that if $p \geq 4$, or $p = 3$ and $k \geq 2$, then MAX $(\Delta \leq k)$-O_p-OVERLAY is NP-complete. We prove it by a double induction on k and p. Theorem 21 and Theorem 22 first prove the base cases of the induction and Lemma 23 corresponds to the inductive steps.

Theorem 21. $(\Delta \leq 1)$-O_4-OVERLAY *is* NP*-complete.*

Proof. We reduce 3-SAT to $(\Delta \leq 1)$-O_4-OVERLAY.

We shall need the following gadget hypergraph $J = J(A, B)$. $V(J) = A \cup B$ with $|A| = 3$ and $|B| = 4$ and $E(J) = \{A \cup \{b\} \mid b \in B\}$.

Claim 21.1. *Every $(O_4, J, 1)$-graph contains exactly one edge with both endvertices in A.*

Proof of claim: Let G be an $(O_4, J, 1)$-graph. Each hyperedge of J contains an edge of G, which necessarily has an endvertex in A. Since $\Delta(G) \leq 1$, G has at most three edges with an endvertex in A. Since J has four hyperedges, there is an edge of G contained in two hyperedges of J. The two endvertices of this edge are in A. $\triangleleft$

Let Φ be an instance of 3-SAT with m clauses $C_1, \ldots, C_m$ on n variables $x_1, \ldots, x_n$. Let $H = H(\Phi)$ be the 4-uniform hypergraph constructed as follows (see Figure 2.3).

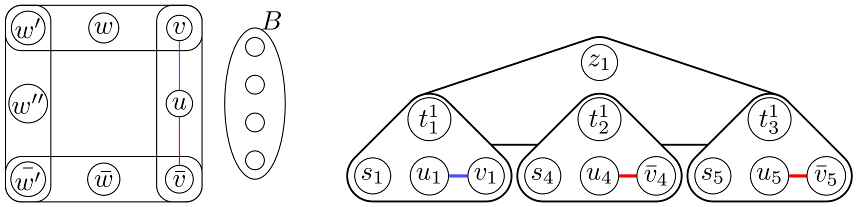


Figure 2.3: **The construction in the reduction.** In a variable gadget (left), a blue (resp. red) edge corresponds to that the variable is $true$ (resp. $false$). And an example of clause gadget of $C_1 = (x_1 \vee \bar{x}_4 \vee \bar{x}_5)$ (right) with three edges $u_1v_1, u_4\bar{v}_4, u_5\bar{v}_5$ which corresponds to an assignment ϕ such that $\phi(x_1) = \phi(\bar{x}_4) = \phi(\bar{x}_5) = true$.

- For each variable x_i, we create the variable gadget VG_i as follows.
  Its vertex set is $\{u_i, v_i, \bar{v}_i, w_i, \bar{w}_i, w_i', \bar{w}_i', u_i'\} \cup B_i$, with $|B_i| = 4$. We add the hyperedges of $J(\{u_i, v_i, \bar{v}_i\}, B_i)$, $J(\{v_i, w_i, w_i'\}, B_i)$, $J(\{\bar{v}_i, \bar{w}_i, \bar{w}_i'\}, B_i)$, and $J(\{u_i', w_i', \bar{w}_i'\}, B_i)$.
- For each clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$, we create the clause hypergraph CG_j. Its vertex set is $\{z_j\} \cup \bigcup_{k=1}^{3} \{s_j^k, t_j^k, u_j^k, v_j^k\}$. Set $Z_j = \{t_j^1, t_j^2, t_j^3, z_j\}$ and for $k \in \{1, 2, 3\}$, set $T_j^k = \{s_j^k, t_j^k, u_j^k, v_j^k\}$. The hyperedge set of CG_j is $\{Z_j, T_j^1, T_j^2, T_j^3\}$. Then for each $k \in \{1, 2, 3\}$, we identify u_j^k and v_j^k with u_i and v_i if $\ell_k = x_i$ and with u_i and $\bar{v}_i$ if $\ell_k = \bar{x}_i$.

Let us prove that there is an $(O_4, H, 1)$-graph if and only if Φ is satisfiable. The general idea of the proof is that the variable gadget enforces only one of edges u_iv_i, $u_i\bar{v}_i$ to belongs G and that $u_iv_i \in E(G)$ (resp. $u_i\bar{v}_i \in E(G)$) if and only if the corresponding variable x_i is $true$ (resp. $false$), thus the appearance of such an edge in a clause gadget implies that there is a true literal in the clause.

Assume first that there is an $(O_4, H, 1)$-graph G.

Claim 21.2. *For each $1 \leq i \leq n$, G contains either the edge u_iv_i or the edge $u_i\bar{v}_i$. Moreover, for each $x \in \{u_i, v_i, \bar{v}_i\}$ there is some $x' \in VG_i$ such that x and x' are adjacent in G.*

Proof of claim: Since VG_i contains $J(\{u_i', w_i', \bar{w}_i'\}, B_i)$, by Claim 21.1, G contains an edge e'' with both endvertices in $\{u_i', w_i', \bar{w}_i'\}$. Assume that e'' is incident to w_i'. Now VG_i contains $J(\{v_i, w_i, w_i'\}, B_i)$. So, by Claim 21.1, G contains an edge e' with both endvertices in $\{v_i, w_i, w_i'\}$. Since $\Delta(G) \leq 1$, e' is not incident to w_i', so $e' = v_iw_i$. Now VG_i contains $J(\{u_i, v_i, \bar{v}_i\}, B_i)$. So, by Claim 21.1, G contains an edge e with both endvertices in $\{u_i, v_i, \bar{v}_i\}$. This edge is not incident to v_i, so $e = u_i\bar{v}_i$. Similarly, we get that G contains $\bar{v}_i\bar{w}_i$ and u_iv_i if e'' is incident to $\bar{w}_i'$. $\triangleleft$

In view of this claim, one defines the truth assignment ϕ by $\phi(x_i) = true$ if $u_iv_i \in E(G)$ and $\phi(x_i) = false$ if $u_i\bar{v}_i \in E(G)$. Let us prove that it satisfies Φ. Consider a clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$ and its corresponding clause gadget CG_j. If there is none of $u_j^kv_j^k$ in $E(G)$ (for all $k \in \{1, 2, 3\}$), that means both u_j^k and v_j^k are incident to an edge which is incident to a vertex in the variable gadget it they belongs to, T_j^k must be O_4-overlaid by the edge $s_j^k t_j^k$. So, t_j^k (for all $k \in \{1,2,3\}$) has degree one in G. However, the hyperedge Z_j is O_4-overlaid by $z_j t_j^k$ or $t_j^k t_j^{k'}$ for $k' \neq k$ and $k, k' \in \{1,2,3\}$, which implies that there is one of vertices t_j^k which is of degree one in $G[Z_j]$. It is a contradiction, thus there is an edge $u_j^k v_j^k \in E(G)$ (for some $k \in \{1,2,3\}$). Hence $\phi(\ell_k) = true$. Hence ϕ satisfies Φ.

Conversely, assume ϕ is an assignment satisfying Φ. Let G be the graph with vertex set $V(H)$ and whose edge set is constructed as follows. For every $1 \leq i \leq n$, add $\{u_i v_i, w_i w_i', u_i' \bar{w}_i', \bar{w}_i \bar{v}_i\}$ to $E(G)$ if $\phi(x_i) = true$ and $\{u_i \bar{v}_i, \bar{w}_i \bar{w}_i', u_i' w_i', w_i v_i\}$ to $E(G)$ if $\phi(x_i) = false$. For every clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$, there exists $k \in \{1,2,3\}$ such that $\phi(\ell_k) = true$. Add the edges $z_j t_j^k$, $s_j^{k+1} t_j^{k+1}$ and $s_j^{k+2} t_j^{k+2}$ (the superscript are modulo 3) to $E(G)$. One easily checks that G is an $(O_4, H, 1)$-graph. □

Theorem 22. *$(\Delta \leq 2)$-O_3-*Overlay *is* NP*-complete.*

Proof. In order to simplify the proof, we will prove that the problem $(\Delta \leq 2)$-O_3-Overlay with handicap is NP-complete, where, in addition to a hypergraph H, we are given a subset of vertices $A \subseteq V(H)$ (called *handicap vertices*), and we ask for a graph G that O_3-overlays H with the additional constraint that for every $v \in A$, the degree of v is at most 1. Observe that we can easily reduce this problem to $(\Delta \leq 2)$-O_3-Overlay: for every $v \in A$, add six vertices $t_v^1, \ldots, t_v^6$, and hyperedges $\{v, t_v^1, t_v^p\}$ for every $p \in \{2,3,4,5,6\}$. Let H' be the obtained hypergraph. Observe that an $(O_3, H, 2)$-graph G in which every $v \in A$ has degree at most 1 can be extended into an $(O_3, H', 2)$-graph by adding the edge vt_v^1 (and all vertices of $V(H') \setminus V(H)$). Conversely, $(O_3, H', 2)$-graph G' must contain the edge vt_v^1, and thus v is of degree at most 1 in $G'[V(H)]$.

We reduce 3-Sat to $(\Delta \leq 2)$-O_3-Overlay with handicap. Let Φ be an instance of 3-Sat with m clauses $C_1, \ldots, C_m$ on n variables $x_1, \ldots, x_n$. We construct a 3-uniform hypergraph H as follows (see Figure 2.4 for an example).

- For every variable x_i, we create a variable gadget VG_i composed of vertices $a_i^1, \ldots, a_i^{p_i}$, $\alpha_i^1, \ldots, \alpha_i^{p_i}$, $b_i^1, \ldots, b_i^{p_i}$, $\beta_i^1, \ldots, \beta_i^{p_i}$, $z_i^1, \ldots, z_i^{p_i}$, and $\zeta_i^1, \ldots, \zeta_i^{p_i}$ where p_i is the number of clauses where x_i appears. Then, for every $j \in [p_i]$, we create the following hyperedges (additions in superscripts are modulo p_i, and we write p_i instead of 0): $h_{i,j}^1 = \{z_i^j, a_i^j, \alpha_i^j\}$, $h_{i,j}^2 = \{z_i^j, b_i^j, \beta_i^j\}$, $h_{i,j}^3 = \{a_i^j, z_i^j, b_i^j\}$, $h_{i,j}^4 = \{z_i^j, b_i^j, \zeta_i^j\}$, $h_{i,j}^5 = \{b_i^j, \zeta_i^j, a_i^{j+1}\}$, $h_{i,j}^6 = \{\zeta_i^j, a_i^{j+1}, z_i^{j+1}\}$.
Moreover, all vertices $z_i^1, \ldots, z_i^{p_i}$, and $\zeta_i^1, \ldots, \zeta_i^{p_i}$ are handicap vertices.
- For every clause C_j, we create a clause gadget CG_j composed of vertices $c_j^1, \ldots, c_j^5$, and, for every $p \in [5]$, of the hyperedge $\{c_j^p, c_j^{p+1}, c_j^{p+3}\}$ (additions are modulo 5, and we write 5 instead of 0). The vertices c_j^4 and c_j^5 are handicap vertices.

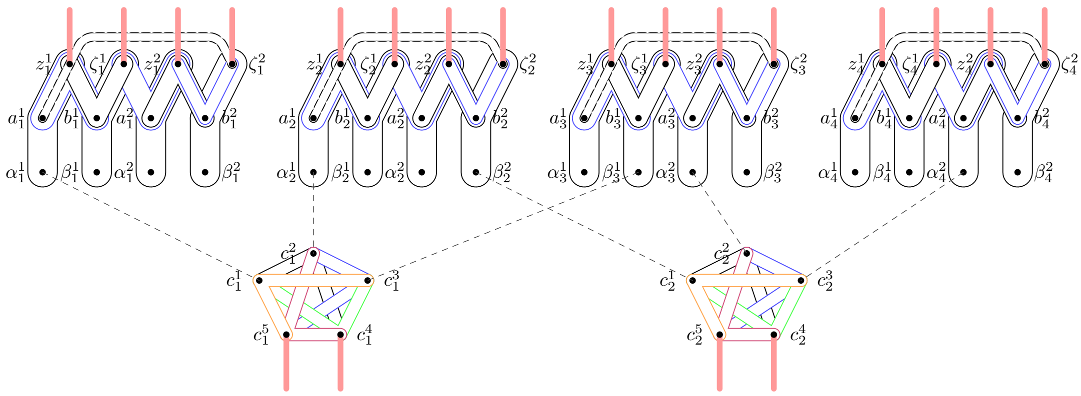


Figure 2.4: **An example of the reduction for $\Phi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4)$.** The dotted lines join two vertices that are identified. The thick red lines indicate the handicap vertices.

- Finally, for every clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$, if $\ell_p = x_i$ (resp. $\ell_p = \bar{x}_i$) for some $p \in \{1, 2, 3\}$, then we identify the vertex c_j^p of CG_j with the vertex α_i^j (resp. β_i^j) of VG_i.

The general idea of the proof is the following. There are only two possibilities to overlay a variable gadget VG_i (either by a set T_i or a set F_i of edges) (see Claim 22.1). This two possibilities correspond to the two possible values ($true$ or $false$) of the variable x_i. Then a clause gagdet CG_j imposes that not all literals of the clauses are assigned $false$ with this correspondence.

Assume first that there is an $(O_3, H, 2)$-graph G such that all handicap vertices have degree at most 1.

For $i \in [n]$, let $T_i = \bigcup_{j=1}^{p_i} \{z_i^j a_i^j, b_i^j \beta_i^j, b_i^j \zeta_i^j\}$ and $F_i = \bigcup_{j=1}^{p_i} \{a_i^j \alpha_i^j, z_i^j b_i^j, \zeta_i^j a_i^{j+1}\}$.

Claim 22.1. *Let $i \in [n]$. We have the following:*

(i) $z_i^j \zeta_i^j \notin E(G)$ and $\zeta_i^j z_i^{j+1} \notin E(G)$, $j \in [m]$.

(ii) Either $T_i \subseteq E(G)$ or $F_i \subseteq E(G)$.

Proof of claim: (i) Assume for a contradiction that $z_i^j \zeta_i^j \in E(G)$. Since z_i^j and ζ_i^j are handicap vertices, the hyperedges $h_{i,j}^2$,$h_{i,j}^3$, $h_{i,j}^5$ must be O_3-overlaid by the edges $b_i^j \beta_i^j$, $a_i^j b_i^j$, $b_i^j a_i^{j+1}$, respectively, contradicting the fact that b_i^j has degree 2 in G. Similarly, if $\zeta_i^j z_i^{j+1} \in E(G)$, then the hyperedges $h_{i,j}^5$,$h_{i,j+1}^1$, $h_{i,j+1}^3$ must be O_3-overlaid by the edges $b_i^j a_i^{j+1}$, $a_i^{j+1} \alpha_i^j$, $a_i^{j+1} b_i^{j+1}$, respectively, a contradiction.

(ii) By (i) the hyperedge $h_{i,j}^4$ is O_3-overlaid by $z_i^j b_i^j$ or $\zeta_i^j b_i^j$.

In the first case $z_i^j b_i^j \in E(G)$, since z_i^j is a handicap vertex, the hyperedge $h_{i,j-1}^6$ must be O_3-overlaid by $\zeta_i^{j-1} a_i^j$, and so hyperedge $h_{i,j-1}^3$ is O_3-overlaid by $z_i^{j-1} b_i^{j-1}$. And by induction, we get that $\bigcup_{j=1}^m \{\zeta_i^j a_i^{j+1}, z_i^j b_i^j\} \subseteq E(G)$. Since the z_i^j are handicap vertices, the hyperedge $h_{i,j}^1$ is O_3-overlaid by $a_i^j \alpha_i^j$. Hence $F_i \subseteq E(G)$.

In the second case $\zeta_i^j b_i^j \in E(G)$, a symmetrical argument shows $T_i \subseteq E(G)$. ◁

We define the truth assignment ϕ by $\phi(x_i) = true$ if $T_i \subseteq E(G)$ and $\phi(x_i) = false$ if $F_i \subseteq E(G)$. This is well-defined by Claim 22.1 and the fact that $E(G)$ cannot contain both T_i and F_i because it has maximum degree 2.

Now, for any $j \in [m]$, there exists $p \in \{1, 2, 3\}$ such that c_j^p has degree 2 in $G[CG_j]$ (If not, then there are at most 2 edges. Hence there is a vertex c_j^p of degree 0, and thus there is a hyperedge containing c_j^p which is not overlaid). Therefore, it has degree 0 in the variable gadget c_j^p belongs to. By our definition of ϕ, we get that $\phi(\ell_p) = true$. Hence ϕ satisfies Φ.

Suppose now that $\phi : \{x_1, \ldots, x_n\} \to \{true, false\}$ is an assignment satisfying Φ. Let us first select a set of edges in the variable gadgets. For every $i \in [n]$, if $\phi(x_i) = true$, then select edges $\{z_i^j, a_i^j\}$, $\{b_i^j, \beta_i^j\}$ and $\{\zeta_i^j, b_i^j\}$ for all $j \in [p_i]$. If $\phi(x_i) = false$, then select edges $\{z_i^j, b_i^j\}$, $\{a_i^j, \alpha_i^j\}$ and $\{\zeta_i^j, a_i^{j+1}\}$ for all $j \in [p_i]$ (additions are modulo p_i). Let E_1 be the set of selected edges, and $G_1 = (V(H), E_1)$. Observe that G_1 O_3-overlays the hyperedges of all variable gadgets: for every $i \in [n]$ and every $j \in [p_i]$,

- $h_{i,j}^1$ contains the edge $\{z_i^j, a_i^j\}$ if $\phi(x_i) = true$, and the edge $\{a_i^j, \alpha_i^j\}$ otherwise.
- $h_{i,j}^2$ contains the edge $\{b_i^j, \beta_i^j\}$ if $\phi(x_i) = true$, and the edge $\{z_i^j, b_i^j\}$ otherwise.
- $h_{i,j}^3$ contains the edge $\{z_i^j, a_i^j\}$ if $\phi(x_i) = true$, and the edge $\{z_i^j, b_i^j\}$ otherwise.
- $h_{i,j}^4$ contains the edge $\{\zeta_i^j, b_i^j\}$ if $\phi(x_i) = true$, and the edge $\{z_i^j, b_i^j\}$ otherwise.
- $h_{i,j}^5$ contains the edge $\{\zeta_i^j, b_i^j\}$ if $\phi(x_i) = true$, and the edge $\{\zeta_i^j, a_i^{j+1}\}$ otherwise.
- $h_{i,j}^6$ contains the edge $\{z_i^{j+1}, a_i^{j+1}\}$ if $\phi(x_i) = true$, and the edge $\{\zeta_i^j, a_i^{j+1}\}$ otherwise.

Moreover the vertices $z_i^1, \ldots, z_i^{p_i}$, and $\zeta_i^1, \ldots, \zeta_i^{p_i}$ have degree 1 in G_1 and the vertices α_i^j have degree 0 in G_1 if and only if $\phi(x_i) = true$ and the vertices β_i^j have degree 0 in G_1 if and only if $\phi(x_i) = false$.

We now select a set of edges in the clause gadgets for every clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$. Since ϕ satisfies Φ, there exists $p = p_j \in \{1, 2, 3\}$ such that $\phi(\ell_p) = true$. Select the edges $\{c_j^p, c_j^{p+2}\}$, $\{c_j^p, c_j^{p+3}\}$ and $\{c_j^{p+1}, c_j^{p+4}\}$. Let E_2 be the set of selected edges in clause gadgets, and $G_2 = (V(H), E_2)$. Clearly, G_2 O_3-overlays all hyperedges of the clause gadgets and the vertices c_j^4 and c_j^5, $j \in [m]$, have degree 1 in G_2. Moreover, for all $j \in [m]$ the vertex $c_j^{p_j}$ is identified with the vertex α_i^j or β_i^j, which has degree 0 in G_1 since $\phi(\ell_p) = true$. Thus $G := (V(H), E_1 \cup E_2)$ has maximum degree 2, which proves that it is an $(O_3, H, 2)$-graph. □

Lemma 23. *If* $(\Delta \leq k)$-O_p-OVERLAY *is* NP-*complete, then* $(\Delta \leq k)$-O_{p+1}-OVERLAY *and* $(\Delta \leq k+1)$-O_p-OVERLAY *are* NP-*complete.*

Let k and p be two positive integers with $p \geq 2$, let $H_{k,p}$ be the p-uniform hypergraph on $(k+1)(p-1)$ vertices whose hyperedge set contains all possible p-subsets of vertices.

We first need the following claim in order to prove Lemma 23.

Claim 23.1. *Every $(O_p, H_{k,p}, k)$-graph is k-regular. Moreover, there exists an $(O_p, H_{k,p}, k)$-graph.*

Proof of claim: Let G be an $(O_p, H_{k,p}, k)$-graph. Assume for a contradiction that it has a vertex v of degree at most $k-1$. Then, observe that $|V(G) \setminus N[v]| \geq (k+1)(p-2)+1$. Now, by removing successively the closed neighborhood of a vertex of the graph (chosen arbitrarily) until the graph is anticomplete, we can construct, together with v, an independent set S of G of size p. This is impossible, since S is a hyperedge of $H_{k,p}$ which is not O_p-overlaid by G.
Finally, observe that the disjoint union of $p-1$ cliques of size $k+1$ is a k-regular graph which O_p-overlays $H_{k,p}$. $\triangleleft$

Proof of Lemma 23. First, we will reduce from $(\Delta \leq k)$-O_p-OVERLAY to $(\Delta \leq k)$-O_{p+1}-OVERLAY. Let H be a p-uniform hypergraph. We construct a $(p+1)$-uniform hypergraph H^* from a copy of H and replace, for every hyperedge S, a hyperedge $S' = S \cup \{v_s\}$. We also add, for every hyperedge S, a set of $(k+1)p-1$ new vertices N_S, and form, together with v_S, a hypergraph H_S isomorphic to $H_{k,p+1}$.

Let G be an (O_p, H, k)-graph. Construct G^* from G by adding, for every hyperedge S of H, a disjoint union of p cliques of size $k+1$ on the vertices $N_S \cup \{v_S\}$. Clearly, G^* is an (O_{p+1}, H^*, k)-graph.

Conversely, if G^* is an (O_{p+1}, H^*, k)-graph, then, by Claim 23.1, v_S must have degree k in N_S for every hyperedge S of H. Thus, the restriction of G^* to the vertices of H is an (O_p, H, k)-graph.

Now, we reduce $(\Delta \leq k)$-O_p-OVERLAY to $(\Delta \leq k+1)$-O_p-OVERLAY.
Let H be a p-uniform hypergraph. Let H' be the p-uniform hypergraph obtained from H by doing the following for each vertex $v \in V(H)$:

- add the vertices $x^v, y_1^v, \dots, y_{p-2}^v$ and the hyperedge $S_v = \{v, x^v, y_1^v, \dots, y_{p-2}^v\}$.
- for any $i = \{1, \dots, p-2\}$, add a copy H_i^v of $H_{k+1,p}$ whose vertices are y_i^v and $(p-1)(k+2)-1$ new vertices.

Assume that G' is an $(O_p, H', k+1)$-graph. By Claim 23.1, the graph induced by G' on $V(H_i^v)$ is $(k+1)$-regular, for all $v \in V(H)$, $1 \leq i \leq p-2$. Therefore each y_i^v has $k+1$-neighbors outside the hyperedge S_v. Hence G' must contain the edge vx^v because it O_p-overlays S_v. Thus the subgraph $G = G'[V(H)]$ has maximum degree at most $\Delta(G') - 1 \leq k$ and G O_p-overlays H. In other words, G is an (O_p, H, k)-graph.

Conversely, assume G is an (O_p, H, k)-graph. Let G' be the graph with vertex set $V(H')$ and whose edge set is the union of $E(G)$, $\{vx^v \mid v \in V(H)\}$, and the edge set of an $(O_p, H_v^i, k+1)$-graph (which exists by Claim 23.1), for all $v \in V(H), 1 \leq i \leq p-2$. One easily checks that G' is $(O_p, H', k+1)$-graph. Therefore, there is an (O_p, H, k)-graph if and only if there is an $(O_p, H', k+1)$-graph. $\square$

Propositions 19 and 20, Theorems 21 and 22, and Lemma 23 imply Theorem 18.

Chapter 3

MAX $(\Delta \leq k)$-F-OVERLAY

In this chapter, we address the complexity of MAX $(\Delta \leq k)$-F-OVERLAY and prove the following theorem.

Theorem 24. MAX $(\Delta \leq k)$-F-OVERLAY *is polynomial-time solvable if either* $\Delta(F) > k$, *or* F *is an anticomplete graph, or* $F = O_2$, *or* $k = 1$ *and* $F = O_3$. *Otherwise it is* NP-*complete.*

3.1 Proof overview

Polynomial cases. As noticed in Section 2.3, if $\Delta(F) > k$ or F is an anticomplete graph then MAX $(\Delta \leq k)$-F-OVERLAY is trivially polynomial-time solvable.
Moreover, by Propositions 19 and 20, MAX $(\Delta \leq 1)$-O_3-OVERLAY and MAX $(\Delta \leq k)$-O_2-OVERLAY (for all positive integers k) are also polynomial-time solvable.

NP**-completeness.** It suffices now to prove that MAX $(\Delta \leq k)$-F-OVERLAY is NP-complete when $\Delta(F) \leq k$, $F \neq \overline{K}_{|F|}$, $|F| \geq 3$ and $(F, k) \neq (O_3, 1)$.
Let F' be the graph induced by the non-isolated vertices of F. Then $F = F' + \overline{K}_q$ with $q = |F| - |F'|$. If $|F'| = 2$, then $F = O_{|F|}$, and we have the result by Theorem 18. If $|F'| \geq 3$, then we can conclude using Theorem 25 and Lemma 26 which are described in the next section.

3.2 NP-completeness

We first establish the NP-completeness when F has no isolated vertices.

Theorem 25. *Let F be a graph on at least three vertices with no isolated vertices. If $k \geq \Delta(F)$, then* MAX ($\Delta \leq k$)-F-OVERLAY *is* NP*-complete on neat hypergraphs.*

Proof. Assume $k \geq \Delta(F)$. Let $n = |F|$, $a_1, \dots, a_n$ be an ordering of the vertices of F such that $\delta(F) = d(a_1) \leq d(a_2) \leq \cdots \leq d(a_n) = \Delta(F)$.
Let $\gamma = \lfloor k/\delta(F) \rfloor - 1$, $\beta = k - \gamma\delta(F)$. Observe that $\delta(F) \leq \beta \leq 2\delta(F) - 1$.

We shall give a reduction from INDEPENDENT SET for cubic graphs [Kar72]. We distinguish two cases depending on whether $d(a_2) > \beta$ or not. Let $\gamma_2 = \lfloor (k - d(a_2))/\delta(F) \rfloor$ and $\gamma_3 = \lfloor (k - d(a_3))/\delta(F) \rfloor$. And we define the value γ_2 as follows.

- If $d(a_2) > \beta$, let $\gamma_1 = \gamma_2$.
- If $d(a_2) \leq \beta$, let $\gamma_1 = \lfloor (k - d(a_1))/\delta(F) \rfloor$.

Let Γ be a cubic graph. For each vertex $v \in V(\Gamma)$, let $(e_1(v), e_2(v), e_3(v))$ be an ordering of the edges incident to v. We shall construct the neat hypergraph $H = H(\Gamma)$ as follows.

- For each vertex $v \in \Gamma$, we create a hyperedge $S_v = \{a_1^v, \dots, a_n^v\}$. Then, for $1 \leq i \leq 3$, we add γ_i a_i^v*-leaves*, that are hyperedges containing a_i^v and $n-1$ new vertices.
- For each edge $e = uv \in \Gamma$, let i and j be the indices such that $e = e_i(u) = e_j(v)$. We create a new vertex z_e and hyperedges S_u^e (S_v^e) containing z_e, a_i^u (a_j^v), and $n-2$ new vertices, respectively. Then, we add γ z_e*-leaves*, that are hyperedges containing z_e and $n-1$ new vertices.

From the construction, observe that γ_i will be useful to enforce the degree of vertex a_i^v for $v \in V(G)$ which could have degree at most $d(a_3)$ in the hypergraph. We shall prove that $over_F(H,k) = (\gamma_1 + \gamma_2 + \gamma_3)|V(\Gamma)| + (\gamma+1)|E(\Gamma)| + \alpha(\Gamma)$, where $\alpha(\Gamma)$ denotes the cardinality of a maximum independent set in Γ.

The following claim shows that there are optimal solutions with specific structure. This leads to the inequality:

$$over_F(H,k) \leq (\gamma_1 + \gamma_2 + \gamma_3)|V(\Gamma)| + (\gamma+1)|E(\Gamma)| + \alpha(\Gamma)$$

Claim 25.1. *There is a graph G with $\Delta(G) \leq k$ that F-overlays $over_F(H,k)$ hyperedges of H such that:*

(a) *each x-leaf L is F-overlaid and x is incident to $\delta(F)$ edges in $G[L]$ (with $x = a_i^v$ or $x = z_e$).*

(b) *for each edge $e = uv \in E(\Gamma)$, exactly one of the two hyperedges S_u^e and S_v^e is F-overlaid. Moreover if S_u^e (S_v^e) is F-overlaid, then a_i^u (a_j^v) is incident to at most $d(a_2)$ edges in S_u^e (S_v^e), respectively.*

(c) *the set of vertices v such that S_v is F-overlaid is an independent set in Γ.*

Proof of claim: (a) Let G be a graph with $\Delta(G) \leq k$ that F-overlays $over_F(H,k)$ hyperedges of H. Observe first that since H is a neat hypergraph, we may assume that, for every hyperedge S of H, $G[S] = F$ if S is F-overlaid and $G[S]$ has no edge otherwise.

Therefore if an x-leaf L is F-overlaid, then replacing $G[L]$ by a copy of F in which x has degree $\delta(F)$, we obtain another graph with maximum degree at most k that F-overlays $over_F(H,k)$ hyperedges of H. Henceforth we may assume that if an x-leaf is F-overlaid then x is incident to $\delta(F)$ edges included in it.

Now assume that the z_e-leaf L, for $e \in E(\Gamma)$, is not F-overlaid by G. We have $d_G(z_e) > k - \delta(F)$ for otherwise adding on L the edges of a copy of F in which z_e has degree $\delta(F)$, we obtain a graph F-overlaying more hyperedges than G. But since there are γ z_e-leaves and z_e is incident to $\delta(F)$ edges in each of them, at least one of the hyperedges S_u^e and S_v^e is F-overlaid. Now replacing the edges of G in this hyperedge by those of a copy of F in L, in which z_e has degree $\delta(F)$, we obtain another graph with maximum degree at most k that F-overlays $over_F(H,k)$ hyperedges of H. Henceforth we may assume that every z_e-leaf is F-overlaid.

A similar reasoning shows that every a_i^v-leaf is F-overlaid, for all $v \in V(\Gamma)$, $i \in \{1,2,3\}$.

(b) Now by (a) each vertex z_e is incident to $\gamma\delta(F) = k - \beta > k - 2\delta(F)$ edges in the z_e-leaves. Thus z_e is incident to at most β edges in $S_u^e \cup S_v^e$. Since $\beta \leq 2\delta(F) - 1$, at most one of S_u^e and S_v^e is F-overlaid.
Moreover, if S_u^e is F-overlaid and $\beta < d(a_2)$ (resp. $d(a_2) \leq \beta$), then z_e must have degree $d(a_1)$ (resp. at most $d(a_2)$) in $G[S_u^e]$.
Assume now that none of S_u^e, S_v^e is F-overlaid, and let $e = e_i(u)$, a_i^u has γ_i a_i^u-leaves, so a_i^u is incident to $\gamma_i\delta(F) \leq k - d(a_i)$ edges in these leaves.
Thus, replacing the edges of G in S_u^e by those of a copy of F, in which

- if $d(a_2) > \beta$, we have $d_G(a_i^u) \leq k - d(a_2)$, then z_e has degree $\delta(F)$ and a_i^u has degree $d(a_2)$.
- if $d(a_2) \leq \beta$, since $d_G(a_1^u) \leq k - d(a_1)$, then for $i = 1$, a_i^u must have degree $d(a_1)$, so z_e has degree $d(a_2)$. And for $i = \{2,3\}$, let $d(z_e) = d(a_1)$, $d(a_i^u) = d(a_2)$.

Hence, we obtain another graph with maximum degree at most k that F-overlays $over_F(H,k)$ hyperedges of H. Henceforth, we may assume that at least one of S_u^e and S_v^e is F-overlaid.

(c) Consider now a vertex $v \in V(\Gamma)$ such that S_v is F-overlaid. We distinguish the following cases.

- If $d(a_2) > \beta$, then by (a), vertex a_1^v is incident to $\gamma_1\delta(F)$ edges in the a_1^v-leaves and at least $\delta(F) \in S_v$. Therefore $S_v^{e_1(v)}$ cannot be F-overlaid for otherwise, by (b), a_1^v would have degree at least $\gamma_1\delta(F) + \delta(F) + d(a_2) > k$. Similarly, $S_v^{e_2(v)}$ cannot be F-overlaid.

  Assume now for a contradiction that $S_v^{e_3(v)}$ is F-overlaid. By (a) and (b), a_3^v is incident to $\gamma_3\delta(F)$ edges in the a_3^v-leaves and at least $d(a_2) \in S_v^{e_3(v)}$. Since $\gamma_3\delta(F) + d(a_2) + d(a_3) > k$, a_3^v is incident to less than $d(a_3)$ edges in S_v. Thus, there is a vertex $a \in \{a_1^v, a_2^v\}$ which is incident to at least $d(a_3)$ edges in S_v and thus at least $\gamma_1\delta(F)+d(a_3)$ edges in G. But a_3^v is adjacent to at least $\delta(F)$ edges in S_v so at least $\gamma_3\delta(F)+d(a_2)+\delta(F)$ edges in total. Thus $\gamma_3\delta(F)+d(a_2)+\delta(F) \leq k$, so $\gamma_3 \leq \gamma_1 - 1$. Thus a is adjacent to at least $(\gamma_3+1)\delta(F)+d(a_3) > k$ edges, a contradiction.

- If $d(a_2) \leq \beta$, then by (a), vertex a_1^v is incident to $\gamma_1\delta(F)$ edges in the a_1^v-leaves and at least $\delta(F) \in S_v$. If $S_v^{e_1(v)}$ is F-overlaid, then (b) yields $d(a_i^v) = \gamma_i\delta(F) + \delta(F) + \delta(F) > k$, a contradiction. If $S_v^{e_2(v)}$ is F-overlaid, then $d(a_2^v) = \delta$ in $G[S_v]$. We obtain that $d(a_2^v) = \gamma_2\delta + d(a_2) + \delta \leq k$ which implies that $\gamma_2 \leq \gamma_2 - 1$, a contradiction. If $S_v^{e_3(v)}$ is F-overlaid, then we get a contradiction in the same way as that in the case $d(a_2) > \beta$.

To summarize, if S_v is F-overlaid, then none of $S_v^{e_1(v)}$, $S_v^{e_2(v)}$, $S_v^{e_3(v)}$ is F-overlaid. Together with (b), this implies (c). ◁

Conversely, consider W a maximum independent set of Γ.

Let G be the graph with vertex $V(H)$ which is the union of the following subgraphs :

- for each x-leaf L, we add a copy of F on L in which x has degree $\delta(F)$;
- for each vertex $v \in W$, we add a copy of F on S_v in which a_i^v has degree $d(a_i)$ for all $1 \leq i \leq n$.
- for each edge $e \in E(\Gamma)$, we choose an endvertex u of e such that $u \notin W$, and add a copy of F in which z_e has degree $d(a_1)$ and a_i^u has degree $d(a_2)$ (with i the index such that $e_i(u) = e$) except the case when $d(a_2) \leq \beta$ and $i = 1$, we do that but $d(z_e) = d(a_2)$ and $d(a_1^u) = d(a_1)$ in $G[S_u^{e_1(u)}]$.

It is simple matter to check that $\Delta(G) \leq k$ and that G F-overlays $(\gamma_1+\gamma_2+\gamma_3)|V(\Gamma)| + (\gamma+1)|E(\Gamma)| + \alpha(\Gamma)$ hyperedges of H. Thus $over_F(H,k) \geq (\gamma_1+\gamma_2+\gamma_3)|V(\Gamma)| + (\gamma+1)|E(\Gamma)| + \alpha(\Gamma)$. □

We then investigate the problem when F has isolated vertices. In order to prove it, we exploit the induction method by using Theorem 25 as the base case, and the induction part is the following.

Lemma 26. *Let k be a positive integer, let F be a graph with $\delta(F) \geq 1$, and let q be a non-negative integer. If* MAX $(\Delta \leq k)$-$(F+\overline{K}_q)$-OVERLAY *is* NP*-complete, then* MAX $(\Delta \leq k)$-$(F+\overline{K}_{q+1})$-OVERLAY *is also* NP*-complete.*

Proof. We shall reduce MAX $(\Delta \leq k)$-$(F+\overline{K}_q)$-OVERLAY to MAX $(\Delta \leq k)$-$(F+\overline{K}_{q+1})$-OVERLAY. Set $p = |V(F)|$ and $r = \lfloor k/\delta(F) \rfloor$. Note that $r \geq 1$, for otherwise MAX $(\Delta \leq k)$-$(F+\overline{K}_q)$-OVERLAY would trivially be polynomial-time solvable.

Let H' be a $(p+q)$-uniform hypergraph. Let H be the $(p+q+1)$-uniform hypergraph obtained from H' as follows. Its vertex set is partitioned into $V(H')$, $P_1, \ldots, P_r$, K, and $\{z\}$ with $|P_i| = p-1$ for all $1 \leq i \leq r$ and $|K| = (m+1)(k+1) + kp$

with $m = \max\{|E(H')|, q+1\}$. The hyperedge set of H contains $Q \cup P_i \cup \{z\}$ for all $(q+1)$-subsets Q of K and all $1 \leq i \leq r$, and $S \cup \{z\}$ for all $S \in E(H')$.

The reduction can be performed in polynomial time, because for each hyperedge in H' we construct a polynomial number of hyperedges in H (recall that q is fixed), and each of them is straightforward to obtain. We shall now prove that $over_{F+\overline{K}_{q+1}}(H,k) = over_{F+\overline{K}_q}(H',k) + r\binom{|K|}{q+1}$.

Let G'_0 be a graph with $V(G'_0) = V(H')$ and $\Delta(G'_0) \leq k$ that $(F+\overline{K}_q)$-overlays $over_{F+\overline{K}_q}(H',k)$ hyperedges of H'. For $1 \leq i \leq r$, let F_i be a copy of F on $P_i \cup \{z\}$ in which z has degree $\delta(F)$. Let G_0 be the graph defined by $V(G_0) = V(H)$ and $E(G'_0) \cup \bigcup_{i=1}^{r} E(F_i)$. Clearly, $\Delta(G_0) \leq k$ and G_0 F-overlays all hyperedges of H except hyperedges $S \cup \{z\}$ in which $S \in E(H')$ are not overlaid by G'_0. So $over_{F+\overline{K}_{q+1}}(H,k) \geq over_{F+\overline{K}_q}(H',k) + r\binom{|K|}{q+1}$.

Conversely, let G be a graph with $V(G) = V(H)$ and $\Delta(G) \leq k$ that maximizes the number of $(F+E_{q+1})$-overlaid hyperedges.

Claim 26.1. *In G, vertex z has at least $\delta(F)$ neighbors in each P_i, $1 \leq i \leq r$.*

Proof of claim: Suppose for a contradiction that z has less than $\delta(F)$ neighbors in P_i. Since $\Delta(G) \leq k$, at most kp vertices of K are incident to an edge with an endvertex in $P_i \cup \{z\}$. Thus there is a subset R of K of $(m+1)(k+1)$ vertices non-adjacent to all vertices of $P_i \cup \{z\}$. Now $G[R]$ has maximum degree k and is thus $(k+1)$-colorable. Hence $G[R]$ has an independent set S of size $m+1$. Observe that for each $(q+1)$-subset Q of S, the subgraph $G[Q \cup P_i \cup \{z\}]$ has $q+2$ vertices of degree less than $\delta(F)$ (namely the ones of $Q \cup \{z\}$), and so the hyperedge $Q \cup P_i \cup \{z\}$ is not $(F+\overline{K}_{q+1})$-overlaid. Thus there are at least $m+1 \geq |E(H')|+1$ hyperedges not $(F+\overline{K}_{q+1})$-overlaid by G. This contradicts the maximality of G as G_0 overlays more hyperedges of H. $\triangleleft$

Now z has degree at least $r\delta(F)$ in $\bigcup_{i=1}^{r} P_i$ and so less than $\delta(F)$ in each hyperedge $S \cup \{z\}$ for $S \in E(H')$. Hence if G $(F+\overline{K}_{q+1})$-overlays $S \cup z$, it must also $(F+\overline{K}_q)$-overlay S. Thus $over_{F+\overline{K}_{q+1}}(H,k) \leq over_{F+\overline{K}_q}(H',k) + r\binom{|K|}{q+1}$. $\square$

Chapter 4

$(\Delta \leq k)$-F-OVERLAY

In this chapter, we investigate the complexity of $(\Delta \leq k)$-F-OVERLAY problem. Unlike MAX $(\Delta \leq k)$-F-OVERLAY which is very hard so leads to a simple dichotomy, $(\Delta \leq k)$-F-OVERLAY should have a more complicated one. As a consequence, we first present in Section 4.1 several pairs (F, k) for which $(\Delta \leq k)$-F-OVERLAY is in P. NP-completeness results are then proved in Sections 4.2, 4.3. Through both sections, we aim to prove that for a graph F, which is is neither a complete nor an anticomplete graph, there is $k_0 = k_0(F)$ such that $(\Delta \leq k)$-F-OVERLAY is NP-complete for all $k \geq k_0$. A graph F having this property is called **ultimately NP-complete**.
In Section 4.4, we study the restriction of $(\Delta \leq k)$-F-OVERLAY on neat hypergraphs. This problem is called NEAT $(\Delta \leq k)$-F-OVERLAY.
In the final section, we give some remarks and present some open questions for further research.

4.1 Polynomial results

Recall the dichotomy for graphs O_p in Section 2.4. $(\Delta \leq k)$-O_p-OVERLAY is polynomial-time solvable (if and only if under the assumption P $\neq$ NP) when $p = 2$ or when $p = 3$ and $k = 1$. We now exhibit more pairs (F, k) such that $(\Delta \leq k)$-F-OVERLAY is polynomial-time solvable. Precisely, it is the case for some regular graphs and paths.

4.1.1 Regular graphs

Proposition 27. *For every complete graph K and every positive integer k, $(\Delta \leq k)$-K-OVERLAY is polynomial-time solvable.*

Proof. Observe that a $|V(K)|$-uniform hypergraph H is a positive instance of $(\Delta \leq k)$-K-OVERLAY if and only if $K(H)$ is a (K, H, k)-graph. □

Proposition 28. *For every connected k-regular graph F, $(\Delta \leq k)$-F-*OVERLAY *is polynomial-time solvable.*

Proof. One easily sees that a $|V(F)|$-uniform hypergraph H admits an (F, H, k)-graph if and only if the hyperedges of H are pairwise non-intersecting. □

Let C_4 denote the cycle on 4 vertices. Proposition 28 implies that $(\Delta \leq 2)$-C_4-OVERLAY is polynomial-time solvable. We now show that $(\Delta \leq 3)$-C_4-OVERLAY is also polynomial-time solvable.

Theorem 29. $(\Delta \leq 3)$-C_4-OVERLAY *is polynomial-time solvable.*

Proof. Let H be a 4-uniform hypergraph.

Let us describe an algorithm to decide whether there is a $(C_4, H, 3)$-graph. It is sufficient to do it when H is connected since the disjoint union of the $(C_4, K, 3)$-graphs for connected components K of H is a $(C_4, H, 3)$-graph.

Observe first that if two hyperedges of H intersect in exactly one vertex u, then no such graph exists, since u must have degree 2 in each of the hyperedges if they are C_4-overlaid, and thus degree 4 in total. Therefore if there are two such hyperedges, we return 'No'. At this point we may assume that $|E(H)| \geq 2$ for otherwise we return 'Yes'.

From now on we may assume that two hyperedges either do not intersect, or are adjacent (intersect on at least two vertices).

Claim 29.1. *If two hyperedges S_1 and S_2 intersect on three vertices and there is a $(C_4, H, 3)$-graph G, then $|V(H)| \leq 6$.*

Proof of claim: Assume $S_1 = \{a_1, b, c, d\}$ and $S_2 = \{a_2, b, c, d\}$. Let G be a $(C_4, H, 3)$-graph. Observe that a_1 and a_2 must have two common neighbors, say b, c, in common in $\{b, c, d\}$, for otherwise their unique common vertex would have degree 4 in G. This forces d to be also adjacent to b and c.

Consider a hyperedge S_3 intersecting $S_1 \cup S_2$. Since it is C_4-overlaid by G, at least two edges connect $S_3 \cap (S_1 \cup S_2)$ to $S_3 \setminus (S_1 \cup S_2)$. The endvertices of those edges in $S_1 \cup S_2$ must have degree 2 in $G[S_1 \cup S_2]$. Hence, without loss of generality, there is a vertex e not in $S_1 \cup S_2$ such that either $S_3 = \{a_1, a_2, b, e\}$ and $\{a_1e, a_1e\} \subseteq E(G)$, or $S_3 = \{a_1, b, d, e\}$ and $\{a_1e, de\} \subseteq E(G)$. Now no hyperedge can both intersect $S_1 \cup S_2 \cup S_3$ and contain a vertex not in $S_1 \cup S_2 \cup S_3$, for such a hyperedge must contain either the vertices d, e or a_2, e which are at distance 3 in $G[S_1 \cup S_2 \cup S_3]$. (However there can be more hyperedges contained in $S_1 \cup S_2 \cup S_3$.) Hence $|V(H)| \leq 6$. ◁

In view of Claim 29.1, if there are two hyperedges with three vertices in common, either we return 'No' if $|V(H)| > 6$, or we check all possibilities (or follow the proof of the above claim) to return the correct answer otherwise. Henceforth, we may assume that any two adjacent hyperedges intersect in exactly two vertices.

Let S_1 and S_2 be two adjacent hyperedges, say $S_1 = \{a, b, c, d\}$ and $S_2 = \{c, d, e, f\}$. Note that every $(C_4, H, 3)$-graph contains the edges ab, cd and ef, and that $N(c) \cup N(d) = S_1 \cup S_2$ in this $(C_4, H, 3)$-graph.

Claim 29.2. *If there is another hyperedge than S_1 and S_2 containing c or d, and there is a $(C_4, H, 3)$-graph G, then $|V(H)| \leq 8$.*

Proof of claim: Without loss of generality, we may assume that G contains the cycle (a, b, d, f, e, c, a) and the edge cd. Hence the only possible hyperedges containing c or d and a vertex not in $S_1 \cup S_2$ are $S_3 = \{a, c, e, g\}$ for some $g \notin S_1 \cup S_2$ and $S_4 = \{b, d, f, h\}$ for some $h \notin S_1 \cup S_2$.

If H contains both S_3 and S_4, then G contains the edges ag, eg, bh and hf. If G contains also gh, then $G[S_1 \cup S_2 \cup S_3 \cup S_4]$ is 3-regular, so $G = G[S_1 \cup S_2 \cup S_3 \cup S_4]$. If G does not contain gh, then the only vertices of degree 2 in $G[S_1 \cup S_2 \cup S_3 \cup S_4]$ are g and h, and they are at distance at least 3 in this graph. Thus every hyperedge intersecting $S_1 \cup S_2 \cup S_3 \cup S_4$ is contained in this set, so $|V(H)| = 8$.

Assume now that H is a superset of only one of S_3, S_4. Without loss of generality, we may assume that this is S_3. Hence G also contains the edges ag and eg. If $V(G) \neq S_1 \cup S_2 \cup S_3$, then there is a hyperedge S that intersects $S_1 \cup S_2 \cup S_3$ and that is not contained in $S_1 \cup S_2 \cup S_3$. It does not contain c and d. Hence it must contain one of the vertices a or e, because it intersects each S_i along an edge of G or not at all. Without loss of generality, $a \in S$. Hence $S = \{a, b, i, g\}$ for some vertex i not in $S_1 \cup S_2 \cup S_3$, and G contains the edges bi and ig. Now, as previously, either i and f are adjacent and $G = G[S_1 \cup S_2 \cup S_3 \cup S]$ or they are not adjacent, and every hyperedge intersecting $S_1 \cup S_2 \cup S_3 \cup S$ is contained in this set. In both cases, $|V(H)| = 8$. ◁

We now summarize the algorithm: if $|V(G)| \leq 8$, then we solve the instance by brute force. Otherwise, for every pair of hyperedges S_1, S_2, if their intersection is of size 1 or 3, we answer 'No'. In the remaining cases, if S_1 and S_2 have non-empty intersection, then, they must intersect on two vertices c and d, and these vertices do not belong to any other hyperedges but S_1 and S_2. Hence, if a vertex is in at least 3 hyperedges, then we return ' No'. If not, then every hyperedge intersects at most two hyperedges on two disjoint pairs of vertices. We can then return the $(C_4, H, 3)$-graph obtained as follows. If a hyperedge $S = \{a, b, c, d\}$ intersects either two other hyperedges S_1 and S_2 is $\{a, b\}$ and $\{c, d\}$, or exactly one hyperedge on $\{a, b\}$, then add the edges ab, bc, cd, da to $E(G)$.

Clearly, the above-described algorithm runs in polynomial time. □

4.1.2 Paths

Let $\mathcal{P}$ be the set of all paths. We have the following:

Theorem 30. $(\Delta \leq 2)$-$\mathcal{P}$-OVERLAY *is linear-time solvable.*

Proof. Clearly, if H is not connected, it suffices to solve the problem on each of the components and to return 'No' if the answer is negative for at least one of the components, and 'Yes' otherwise. Henceforth, we shall now assume that H is connected. In such a case, a $(\mathcal{P}, H, 2)$-graph is either a path or a cycle. However, if H is $\mathcal{P}$-overlaid by a path P, then it is also $\mathcal{P}$-overlaid by the cycle obtained from P by adding an edge between its two endvertices. Thus, we focus on the case where G is a cycle.

Let $\mathcal{S}$ be a family of sets. The *intersection graph* of a family $\mathcal{S}$ is the graph $IG(\mathcal{S})$ whose vertices are the sets of $\mathcal{S}$, and in which two vertices are adjacent if the corresponding sets in S intersect.

The *intersection graph* of a hypergraph H, denoted by $IG(H)$, is the intersection graph of its hyperedge set. We define two functions l_H and s_H as follows:

$$l_H(S) = |S| - 1 \text{ for all } S \in E(H) \quad \text{and} \quad s_H(S, S') = |S \cap S'| - 1 \text{ for all } S, S' \in E(H).$$

Let $\mathbb{C}_\ell$ be the circle of circumference ℓ. We identify the points of $\mathbb{C}_\ell$ with the integer numbers (points) of the segment $[0, \ell]$, (with 0 identified with ℓ). A *circular-arc graph* is the intersection graph of a set of arcs on $\mathbb{C}_\ell$. A set $\mathcal{A}$ of arcs such that $IG(\mathcal{A}) = G$ is called an *arc representation* of G. We denote by A_v the arc corresponding to v in $\mathcal{A}$. Let G be a graph and let $l : V(G) \to \mathbb{N}$ and $s : E(G) \to \mathbb{N}$ be two functions. An arc representation $\mathcal{A}$ of G is *l-respecting* if A_v has length $l(v)$ for any $v \in V(G)$, *s-respecting* if $A_v \cap A_u$ has length $s(u, v)$ for all $uv \in E(G)$, and *(l, s)-respecting* if it is both l-respecting and s-respecting. One can easily adapt the algorithm given by Köbler et al. [KKW15] for (l, s)-respecting interval representations to decide in linear time whether a graph admits an (l, s)-respecting arc representation in $\mathbb{C}_n$ for every integer n.

Claim 30.1. *Let H be a connected hypergraph on n vertices. There is a cycle $\mathcal{P}$-overlaying H if and only if $IG(H)$ admits an (l_H, s_H)-respecting arc representation into $\mathbb{C}_n$.*

Proof of claim: Assume that H is $\mathcal{P}$-overlaid by a cycle $C = (v_0, v_1, \ldots, v_{n-1}, v_0)$. There is a canonical embedding of C to $\mathbb{C}_n$ in which every vertex v_i is mapped to i and every edge $v_i v_{i+1}$ to the circular arc $[i, i+1]$. For every hyperedge $S \in E(H)$, $P[S]$ is a subpath, which is mapped to the circular arc A_S of $\mathbb{C}_n$ that is the union of the circular arcs to which its edges are mapped. Clearly, $\mathcal{A} = \{A_S \mid S \in E(H)\}$ is an (l_H, s_H)-respecting interval representation of $IG(H)$.

Conversely, assume that $IG(H)$ admits an (l_H, s_H)-respecting interval representation $\mathcal{A} = \{A_S \mid S \in E(H)\}$ into $\mathbb{C}_n$. Let S_0 be a hyperedge of minimum size. Free

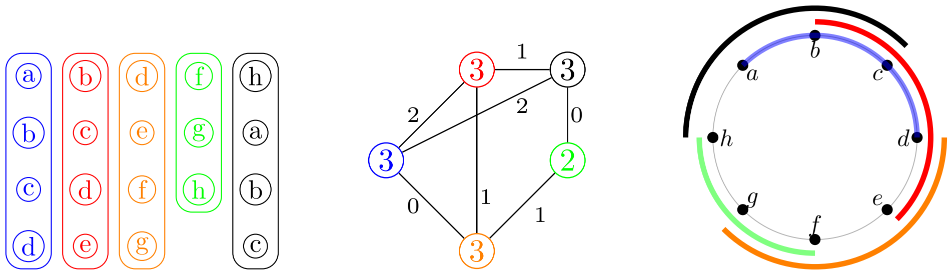


Figure 4.1: **Example of an intersection graph and a corresponding circular representation.** Given a hypergraph H (left), we determine its graph $IG(H)$ (middle) with each vertex represents the hyperedge of the same color in which the value of any vertex $S \in V(H)$ is $l_H(S)$ and the weight of an edge between S, S' is $s_H(S, S')$. And a circular arc representation (each arc is associated to a hyperedge of the same color) is shown in the right which is (l, s)-respecting.

to rotate all intervals, we may assume that A_{S_0} is $[1, |S_0|]$. Now since $\mathcal{A}$ is (l_H, s_H)-respecting and H is connected, we deduce that the extremities of A_S are integers for all $S \in E(H)$. Let v_1 be a vertex of H that belongs to the hyperedges whose corresponding arcs of $\mathcal{A}$ contain 1. Then for all $i = 2$ to $n = |V(H)|$, denote by v_i an arbitrary vertex not in $\{v_1, \ldots, v_{i-1}\}$ that belongs to the hyperedges whose corresponding arcs of $\mathcal{A}$ contain i. Such a vertex exists because $\mathcal{A}$ is (l_H, s_H)-respecting. Observe that such a construction yields $S = \{v_i \mid i \in A_S\}$ for all $S \in E(H)$. Furthermore, the cycle $C = (v_1, \ldots, v_n, v_1)$ $\mathcal{P}$-overlays H. Indeed, for each $S \in E(H)$, $C[S]$ is the subpath corresponding to A_S, that is $V(C[S]) = \{v_i \mid i \in A_S\}$ and $E(C[S]) = \{v_i v_{i+1} \mid [i, i+1] \subseteq A_S\}$. $\triangleleft$

The algorithm to solve $(\Delta \leq 2)$-$\mathcal{P}$-OVERLAY for a connected hypergraph H in linear time is thus the following:

1. Construct the intersection graph $IG(H)$ and compute the associated functions l_H and s_H.
2. Check whether graph $IG(H)$ has an (l_H, s_H)-respecting interval representation. If it is the case, return 'Yes'. If not return 'No'.

$\square$

Remark 31. We can also detect in polynomial time whether a connected hypergraph H is $\mathcal{P}$-overlaid by a path. Indeed, similarly to Claim 30.1, one can show that there is a path $\mathcal{P}$-overlaying H if and only if $IG(H)$ admits an (l_H, s_H)-respecting interval representation.

4.2 Ultimate NP-completeness

The aim of this section is to show that for a fixed graph F, $(\Delta \leq k)$-F-OVERLAY is NP-complete when $k \geq k_0$ (that k_0 depends of F), then it is ultimately NP-complete

(according to k).
We thus define $\mathrm{np}(F)$ the minimum integer k_0 such that $(\Delta \leq k)$-F-OVERLAY is NP-complete for all $k \geq k_0$ or $+\infty$ if no such k_0 exists. Determining the value $\mathrm{np}(F)$ for some F is interesting in order to answer the complexity dichotomy of $(\Delta \leq k)$-F-OVERLAY.

Theorem 32. $\mathrm{np}(F) = +\infty$ *if and only if* F *is complete or anticomplete.*

Let H be a p-uniform hypergraph. The anticomplete graph on $V(H)$ vertices $\overline{K}_p$-overlays H. Thus, for any non-negative integer k, the answer to $(\Delta \leq k)$-$\overline{K}_p$-OVERLAY is always affirmative, and so this problem can be trivially solved in polynomial time. Hence, $\mathrm{np}(\overline{K}_p) = +\infty$ for all positive integer p.

If K_p is complete, then let G be te graph with vertex $V(H)$ in which two vertices are adjacent if and only if they belong to a same hyperedge of H. Obviously, a graph K_p-overlays H if and only if it contains G as a subgraph. Hence, to solve $(\Delta \leq k)$-K_p-OVERLAY it suffices to build G and to check whether $\Delta(G) \leq k$, which can be done in polynomial time. Thus $\mathrm{np}(K_p) = +\infty$ for all positive integer p.

The next is for the remaining graphs.

Theorem 33. *If* F *is neither a complete graph nor an anticomplete graph, then* $\mathrm{np}(F) < +\infty$.

We first show it is sufficient to prove Theorem 33 for F with no isolated vertices. Then we prove in Lemma 40 that for such a graph F, as soon as there exists k such that $(\Delta \leq k)$-F-OVERLAY is NP-complete, then $\mathrm{np}(F) < +\infty$. It then remains to prove the following one.

Theorem 34. *Let* F *be a graph with no isolated vertex and which is not complete. There exists* k *such that* $(\Delta \leq k)$-F-OVERLAY *is* NP*-complete.*

In Section 4.2.2, we first prove Theorem 34 when F is in some particular classes of graphs : F is **regular** (Theorem 41), F is a **complete graph** K **minus an edge**, denoted by K^- (Theorem 42), and F is **a disjoint union of the complete bipartite graph** $K_{a,a+1}$ (Theorem 43). Then, in Section 4.3, we prove Theorem 34 in full. Its proof requires the previously established particular cases and uses the techniques introduced in proving them.

Some notations and definitions

In the previous sections, we obtained the complexity results which for graphs with specific structures. Such structure are an advantage to exploit and provide the proofs. In this section, the result is general and so we cannot look at every graph and use its own structure. Instead, we use some characteristic of graphs, the degree of each vertex in a graph.

The *degree sequence* of a graph F is the non-decreasing sequence $\mathbf{d} = \{d_1, d_2, \dots, d_p\}$ such that there exists an ordering $(v_1, \dots, v_p)$ of the vertices of F such that $d(v_i) = d_i$ for all $i \in [p]$. We denote by $\lambda_1 < \lambda_2 < \cdots < \lambda_t$ the different values of $\mathbf{d}$ (that are the integers λ in which there exists j such that $d_j = \lambda$). We also denote by α_i the *multiplicity* or number of occurrences of λ_i in $\mathbf{d}$: $\alpha_i = |\{j \mid d_j = \lambda_i\}|$. Observe that $d_1 = \lambda_1 = \delta(F)$ and $d_p = \lambda_t = \Delta(F)$.

The *degree* of a vertex v in H, denoted by $d_H(v)$ or simply $d(v)$ if H is clear from the context, is the number of hyperedges of H containing v. In a hypergraph H, a hyperedge S is *pendant* at a vertex x, if $d_H(v) = 1$ for all $v \in S \setminus \{x\}$.

Let F be a graph, H a hypergraph, and G a graph F-overlaying H. For each hyperedge $S \in E(H)$, one can choose a copy F_S of F which is a subgraph of $G[S]$. We then say that v is a λ_i*-vertex* in S or has degree λ_i in S if v has degree λ_i in F_S.

4.2.1 Reduction to Theorem 34

Graphs with isolated vertices

Lemma 35. *Let F be a graph. If $(\Delta \leq k)$-F-*OVERLAY *is* NP*-complete, then $(\Delta \leq k)$-$(F + \overline{K}_1)$-*OVERLAY *is also* NP*-complete.*

Proof. We shall give a reduction from $(\Delta \leq k)$-F-OVERLAY to $(\Delta \leq k)$-$(F + \overline{K}_1)$-OVERLAY.

Let $\mathbf{d}$ be the (non-decreasing) degree sequence of F, and let λ^+ be the first non-zero value in this sequence. ($\lambda^+ = \lambda_1$ if F has no isolated vertex, and $\lambda^+ = \lambda_2$ otherwise.)

Let H be an $|F|$-uniform hypergraph. We construct an $(|F| + 1)$-uniform hypergraph H' as follows.

- Let $H_1, \dots, H_t$ be $t = \lfloor \frac{k}{\lambda^+} \rfloor |E(H)| + 1$ disjoint copies of H. We add $V(H_i)$ to $V(H')$ for all $i \in [t]$.
- For any $S \in E(H)$, we add a new vertex v_S to $V(H')$. For all $i \in [t]$, denoting by S_i the copy of S in H_i, we add the hyperedge $S'_i = S_i \cup \{v_S\}$ to H'.

We shall prove that there is an (F, H, k)-graph G if and only if there exists an $(F + \overline{K}_1, H', k)$-graph G'.

Assume first that there is an (F, H, k)-graph. We build a graph G' by taking $G'[H_i] = G$ for any $i \in [t]$. Observe that $G'[S'_i]$ is $(F + \overline{K}_1)$-overlaid since $G[S_i]$ is F-overlaid and v_S is an isolated vertex. Furthermore G' has at most degree k. Thus, G' is an $(F + \overline{K}_1, H', k)$-graph.

Conversely, assume that there exists an $(F + \overline{K}_1, H', k)$-graph G'. We will prove that there exists a copy H_i of H such that $G'[H_i]$ is an (F, H, k)-graph. Observe that, for any $S \in E(H)$, the vertex v_S is either isolated or has degree at least λ^+ in each $G'[S'_i]$ for $i \in [t]$. Thus, v_S is not a 0-vertex in at most $\lfloor \frac{k}{\lambda^+} \rfloor$ hyperedges. Since there are $|E(H)|$ such vertices, there exists a copy H_i of H such that for any $S \in E(H)$, v_S is a 0-vertex in all hyperedges $G'[S'_i]$. Thus $G'[H_i]$ is an (F, H, k)-graph. □

Applying the lemma several times, we get the following.

Corollary 36. *Let F be a graph and q a positive integer. If $(\Delta \leq k)$-F-Overlay is NP-complete, then $(\Delta \leq k)$-$(F + \overline{K}_q)$-Overlay is also NP-complete. Hence $\mathrm{np}(F + \overline{K}_q) \leq \mathrm{np}(F)$.*

The family of graphs with isolated vertices to which this result does not apply is $K_p + \overline{K}_q$ because $(\Delta \leq k)$-K_p-Overlay is in P. We then need the following.

Theorem 37. $\mathrm{np}(K_p + \overline{K}_1) \leq 2p - 2$ *for all* $p \geq 2$.

Proof. Let $p \geq 2$ and $k \geq 2p - 2$. We shall prove that $(\Delta \leq k)$-$(K_p + \overline{K}_1)$-Overlay is NP-complete.

Let q and r be the integers such that $k = (p-1)q + r$ with $0 \leq r < p - 1$. Note that $q \geq 2$ since $k \geq 2(p-1)$.

We need the following gadget. Let u be a vertex. A $(p-1)$-*gadget at* u is the hypergraph H_u constructed as follows. The vertex set of H_u is the disjoint union of $\{u, v\}$ and $q + 1$ sets $U_1, \dots, U_{q+1}$ of $p - 1$ vertices, and its hyperedges are $\{u, v\} \cup U_i$ for $i \in [q+1]$, see Figure 4.2.

Claim 37.1. *Let H_u be a $(p-1)$-gadget at u.*

(i) *u has degree at least $p - 1$ in every (F, H_u, k)-graph.*
(ii) *There is an (F, H_u, k)-graph in which u has degree $p - 1$.*

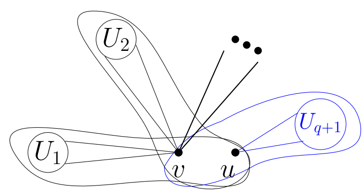


Figure 4.2: **The gadget H_u at vertex u.** An example of an (F, H_u, k)-graph where v has degree $p - 1$ in each of hyperedges $U_1, \dots, U_q$ and u has degree $p - 1$ in U_{q+1}.

Proof of claim: (i) Let G_u be an (F, H_u, k)-graph. Assume for a contradiction that u has degree less than $p - 1$. Then u must be a 0-vertex in each S_i, $i \in [q+1]$. Hence v must be adjacent to the $p - 1$ vertices of U_i in each S_i. Thus v as degree at least $(p-1)(q+1) > k$ in G_u, a contradiction.

(ii) For $i \in Q$, let F_i be a copy of $K_p + \overline{K}_1$ in which u is isolated, and let F_{q+1} be a copy of $K_p + \overline{K}_1$ in which v is isolated, and let $G_u = \bigcup_{ß \in [q+1]} F_i$. Clearly, G_u F-overlays H_u, v has degree $q(p-1) \leq k$ in G_u and u has degree $p - 1$ in G_u. So G_u is the desired (F, H_u, k)-graph. ◁

We give a reduction from 3-COLORING on 4-regular graphs.
Given a 4-regular graph G, we build a p-uniform hypergraph H as follows.

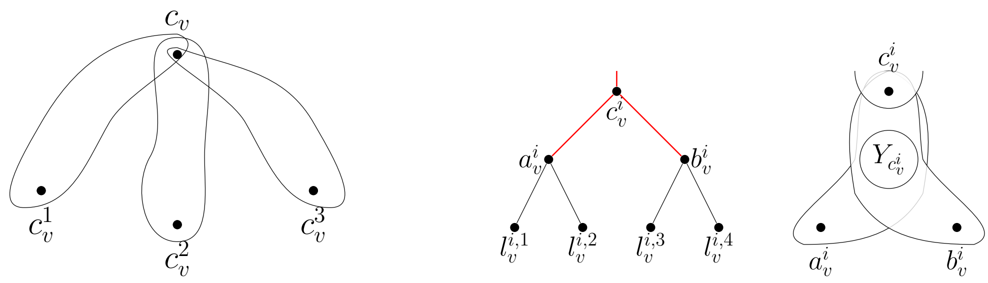


Figure 4.3: **The construction in the reduction.** The vertex gadget VG_v in the left with 3 hyperedges (left) in which at least one of c_v^1, c_v^2, c_v^3 can be a $(p-1)$-vertex in its parent hyperedge. At each of c_v^i, we add to binary tree (middle) where every couple of red edges are replaced by an x-edge-gadget (right) which enforces vertex children having the same degree as its vertex parent in their parent hyperedges. A vertex c_v^i is a $(p-1)$-vertex in its parent hyperedge S_v^i if and only if v has color i in a proper coloring.

- For each vertex $v \in V(G)$, we create a *vertex gadget* VG_v with three hyperedges $S_v^i = \{c_v, c_v^i\} \cup X_v^i$ for $i \in [3]$ where $|X_v^i| = p-2$. We add $q-2$ $(p-1)$-gadgets at c_v. We say that S_v^i is the *parent hyperedge* of c_v^i for each $i \in [3]$.
- For each vertex v and each $i \in [3]$, we construct a *color gadget* CG_v^i for $i \in [3]$ as follows.
  - We create a binary tree T_v^i with vertex set $\{c_v^i, a_v^i, b_v^i, \ell_v^{i,1}, \ell_v^{i,2}, \ell_v^{i,3}, \ell_v^{i,4}\}$ and edge set $\{c_v^i a_v^i, c_v^i b_v^i, a_v^i \ell_v^{i,1}, a_v^i \ell_v^{i,2}, b_v^i \ell_v^{i,3}, b_v^i \ell_v^{i,4}\}$, rooted at c_v^i. In this tree, a_v^i and b_v^i are the children of c_v^i, $\ell_v^{i,1}$ and $\ell_v^{i,2}$ are the children of a_v^i, and $\ell_v^{i,3}$ and $\ell_v^{i,4}$ are the children of b_v^i.
  - For any vertex $x \in \{c_v^i, a_v^i, b_v^i\}$, let y_1, y_2 be its children in T_v^i, and let $e_1 = xy_1, e_2 = xy_2$. We construct an x-edge-gadget as follows: we add a set Y_x of $p-2$ new vertices, the hyperedges $S(e_1) = \{x, y_1\} \cup Y_x$ and $S(e_2) = \{x, y_2\} \cup Y_x$. For convenience, we say that $S(xy_1)$ (resp. $S(xy_2)$) is the *parent hyperedge* of y_1 (resp. y_2). Moreover, for any leaf $\ell_v^{i,j}$, we denote by $S_v^{i,j}$ the hyperedge containing the vertex $\ell_v^{i,j}$. We then add $q-1$ $(p-1)$-gadgets at x.
- For every vertex $v \in V(G)$, let $e_v^1, e_v^2, e_v^3, e_v^4$, be an ordering of the edges incident to v. For each edge $uv \in E(G)$, let j_u and j_v be the indices such that $uv = e_u^{j_u} = e_v^{j_v}$. Then, for all $i \in [3]$, we identify the vertices ℓ_u^{i,j_u} and ℓ_v^{i,j_v} and we add $q-1$ $(p-1)$-gadgets at this vertex.

Let us now prove that there is a proper 3-coloring of G if and only if there is an (F, H, k)-graph G^*.

Assume first that there is an (F, H, k)-graph G^*.

Let $v \in V(G)$. By Claim 37.1 (i), the vertex c_v has degree at least $p-1$ in each of its $(p-1)$-gadgets. Thus it has degree at most $2(p-1)+r$ in $S_v^1 \cup S_v^2 \cup S_v^3$. But those hyperedges pairwise intersect in $\{c_v\}$. Thus there is $i \in [3]$ such that c_v is a 0-vertex in S_v^i. Since we have only one 0-vertex in S_v^i, thus c_v^i must be a $(p-1)$-vertex in S_v^i. Therefore, we can define a 3-coloring ϕ by $\phi(v) = i$ where i is an index such that c_v^i is a $(p-1)$-vertex in S_v^i. Let us now prove that ϕ is proper. We need the following claim.

Claim 37.2. *Let $v \in V(G)$ and $i \in [3]$. If c_v^i is a $(p-1)$-vertex in S_v^i, then so is the leaf $\ell_v^{i,j}$ in $S_v^{i,j}$ for all $j \in [4]$.*

Proof of claim: It suffices to prove that for any $x \in \{c_v^i, b_v^i, a_v^i\}$, if x is a $(p-1)$-vertex in its parent hyperedge, then so are both y_1, y_2 in their parent hyperedges.

Assume that x is a $(p-1)$-vertex in its parent hyperedge. Since x has degree at least $p-1$ in each of its $(p-1)$-gadgets by Claim 37.1 (i), it has degree at most r in $S(xy_1) \cup S(xy_2)$. It implies that x is a 0-vertex in both $S(xy_1), S(xy_2)$. Hence, the vertex y_1 (resp. y_2) must be a $(p-1)$-vertex in $S(xy_1)$ (resp. $S(xy_2)$). $\lhd$

Consider an edge $uv \in E(G)$, $i \in [3]$. By Claim 37.1 (i), the vertex $\ell = \ell_u^{i,j_u} = \ell_v^{i,j_v}$ has degree at least $p-1$ in each of its $q-1$ $(p-1)$-gadgets. Thus it has at most $(p-1)+r$ neighbors in $S_u^{i,j_u} \cup S_v^{i,j_v}$. As ℓ is the unique common vertex of S_u^{i,j_u} and S_v^{i,j_v}, it is a $(p-1)$-vertex in at most one of those. Hence, by the Claim 37.2, at most one of c_u^i, c_v^i is a $(p-1)$-vertex in its parent hyperedge. Thus at most one of u, v is colored i by ϕ. Therefore, ϕ is a proper 3-coloring of G.

Conversely, let ϕ be a proper 3-coloring of G. We construct a graph G^* as follows.

- For any vertex gadget VG_v, $i \in [3]$, let $G^*[S_v^i]$ be a copy of F in which every vertex in X_v^i is a $(p-1)$-vertex, and c_v is a 0-vertex (resp. $(p-1)$-vertex) and c_v^i is a $(p-1)$-vertex (resp. 0-vertex) in S_v^i if $\phi(v) = i$ (resp. $\phi(v) \neq i$).
- In every color gadget CG_v^i, for $x \in \{c_v^i, b_v^i, a_v^i\}$ with children y_1 and y_2, let $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$) be two similar copies of F such that:
  - if $i \neq \phi(v)$, then x has degree $p-1$ in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$; y_1 and y_2 are 0-vertices in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$, respectively (so x has degree $p-1$ in $G^*[S(xy_1) \cup S(xy_2)]$).
  - if $i = \phi(v)$, then x has degree 0 in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$; y_1 and y_2 are $(p-1)$-vertices in $S(xy_1)$ and $S(xy_2)$, respectively (so x has degree at most $p-1$ in $G^*[S(xy_1) \cup S(xy_2)]$).
  - every vertex in Y_x is a $p-1$ vertex in both $S(xy_1)$ and $S(xy_2)$ and so has degree at most p in $G^*[S(xy_1) \cup S(xy_2)]$;
- For any $(p-1)$-gadget H_x at vertex some x, we let $G^*[V(H_x)]$ be an (F, H_x, k)-graph in which v has degree $p-1$. Such a copy exists by Claim 37.1 (ii).

By construction, G^* F-overlays H. Let us check that $\Delta(G^*) \leq k$. Let u be a vertex of G^*.

- If u is in at most two hyperedges (in particular, if u is in X_v^i or u is in Y_x for x internal vertex in some T_v^i or u is only in a $(p-1)$-gadget), then u has degree at most $2(p-1)$, and so at most k.
- Assume now that $u \in \{c_v^i, a_v^i, b_v^i\}$ for $i \in [3]$ with u parent of y_1, y_2. Then u has degree $p-1$ in each of its $q-2$ $(p-1)$-gadgets. Moreover if $i = \phi(v)$ (resp. $i \neq \phi(v)$), then u has degree $p-1$ (resp. 0) in its parent hyperedge and $p-1$ (resp. 0) in $G^*[S^1_{uy_1} \cup S^1_{uy_2}]$. Hence u has degree at most $(q-1)(p-1)+(p-1) = q(p-1) \leq k$.
- Assume that u is the identification of ℓ_v^{i,j_v} and ℓ_w^{i,i_w} for an edge $vw \in E(G)$. First, u has degree $p-1$ in each of its $q-1$ $(p-1)$-gadget. Moreover, since either $\phi(v) \neq i$ or $\phi(w) \neq i$, then u has degree $p-1$ in at most one of S_v^{i,j_v}, S_w^{i,j_w} and 0 in the other. Therefore, u has degree at most $q(p-1) \leq k$ in G^*.

Consequently, G^* is an (F, H, k)-graph. □

Corollary 36 and Theorem 37 directly imply the following.

Corollary 38. *Theorem 33 holds if and only if it holds for graphs with no isolated vertices.*

Reduction to Theorem 34

By Corollary 38, one can restrict our study to graphs F with $\delta(F) \geq 1$. We shall now prove that for such an F, we have $\mathrm{np}(F) \leq +\infty$ as soon as there is some k for which $(\Delta \leq k)$-F-OVERLAY is NP-complete. To prove this, we introduce the notion of degree-gadget that will be useful in almost all the later proofs.

Let F be graph with $\delta(F) \geq 1$. For any integer $d \geq \lambda_1$, a *d-degree-gadget* (with respect to F) at vertex v, is the subgraph $D(d, v)$ defined as follows. Let $\alpha = \lfloor d/\lambda_1 \rfloor$ and $\beta = d - \alpha\lambda_1$. If $\beta = 0$, then $D(d, v)$ is the union of α pendant hyperedges at v. If $\beta \geq 1$, then $D(d, v)$ is the union of $\alpha - 1$ pendant hyperedges at v and two hyperedges which intersect in $I \cup \{v\}$ where I is a set of $\lambda_1 - \beta$ vertices (see Figure 4.4).

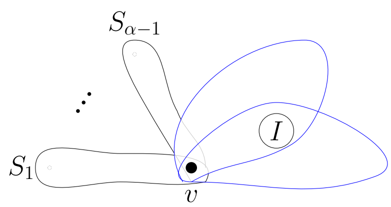


Figure 4.4: **A d-degree-gadget D at vertex v.** The set $I \neq \emptyset$ is the intersection of the two blue hyperedges. $\beta \neq 0$ when I is different from these two hyperedges; and $\beta = 0$ when I and the two blue hyperedges are equal.

Degree-gadgets are useful because of the following proposition whose easy proof is left to readers.

Proposition 39. *Let F be graph with $\delta(F) \geq 1$. Then for any $d \geq \lambda_1$, we have the following.*

(i) *In any graph G that F-overlays $D(d, v)$, vertex v has degree at least d.*

(ii) *There is a graph G_v that F-overlays $D(d, v)$ in which v has degree exactly d, and every other vertex has degree at most $\Delta(F)$ if $\delta(F)$ divides d (i.e. $\beta = 0$) and at most $2\Delta(F) - 1$ otherwise.*

Lemma 40. *Let F be a graph with $\delta(F) \geq 1$. Assume that $(\Delta \leq k_0)$-F-OVERLAY is NP-complete.*

(i) *If $\delta(F) = 1$, then $\mathrm{np}(F) \leq k_0$.*

(ii) $\mathrm{np}(F) \leq \max\{k_0 + \delta(F), 2\Delta(F) - 1\}$.

Proof. Observe that $k_0 \geq \Delta(F)$, because $(\Delta \leq k)$-F-OVERLAY is trivially polynomial-time solvable for every $k < \Delta(F)$.

(i) Let $k > k_0$. We shall prove that $(\Delta \leq k)$-F-OVERLAY is NP-complete. We give a reduction from $(\Delta \leq k_0)$-F-OVERLAY. Let H_0 be an $|F|$-uniform hypergraph. Let H be the hypergraph obtained from H_0 by adding a $(k - k_0)$-degree-gadget DG_v on every vertex v. Such a degree-gadget exists because $k - k_0 \geq 1 = \delta(F)$. Let us prove that there is an (F, H_0, k_0)-graph G_0 if and only if there exists an (F, H, k)-graph G.

Assume there is an (F, H_0, k_0)-graph G_0. By Proposition 39-(ii), for every $v \in V(H_0)$, there is a graph G_v that F-overlays DG_v, in which v has degree $k - k_0$., and every other vertex as degree at most $\Delta(F) \leq k_0$. Consider $G = G_0 \cup \bigcup_{v \in V(H_0)} G_v$. Clearly, G is an (F, H, k)-graph.

Conversely, assume that there is an (F, H, k)-graph G. By Proposition 39-(i), every vertex v of $V(H_0)$ has degree at least $k - k_0$ in DG_v. Thus it has degree at most k_0 in $G[V(H_0)]$. Therefore, $G[V(H_0)]$ is an (F, H_0, k_0)-graph.

(ii) The proof is identical to (i). Taking $k \geq \max\{k_0 + \delta(F), 2\Delta(F) - 1\}$ and using the same reduction as above we get that $(\Delta \leq k)$-F-OVERLAY is NP-complete. Note that $(k - k_0)$-degree-gadgets exist because $k - k_0 \geq \delta(F)$. □

By this lemma, in order to prove that $(\Delta \leq k)$-F-OVERLAY is NP-complete for any $k \geq k_0$ for some k_0, it suffices to prove that $(\Delta \leq k_0)$-F-OVERLAY is NP-complete and thus it is an upper bound of $\mathrm{np}(F)$.

4.2.2 Particular cases

In this section, we prove the NP-completeness of $(\Delta \leq k)$-F-OVERLAY for pairs (F, k) where F is either a **regular graph**, or a **complete graph minus an edge** K^- (*i.e.* it is obtained by removing an edge from K_p) or **a disjoint union of the graph** $K_{a,a+1}$, and k is an integer (depending on F).

Regular graphs

Theorem 41. *Let λ be a positive integer, and let F be a λ-regular graph which is not complete. Then $(\Delta \leq 6\lambda - 1)$-$F$-*OVERLAY *is* NP*-complete.*

Proof. Set $p = |F|$. Since F is not complete, we have $p > \lambda + 1$.

We give a reduction from (3,4)-SAT to $(\Delta \leq 6\lambda - 1)$-$F$-OVERLAY.

Given a formula Φ of (3,4)-SAT with n variables $x_t, t \in [n]$, and m clauses $C_j, j \in [m]$, we construct a p-uniform hypergraph H as follows.

- For each variable x_t, we construct a *variable gadget* H_t as follows. We first create a *center vertex* w_t, a set of $4p - 4$ vertices $U_t = \{u_t^1, \ldots, u_t^{4p-4}\}$, and $4p - 4$ hyperedges $S_t^j = \{w_t, u_t^j, \ldots, u_t^{j+p-2}\}$ (superscripts are modulo 4) for $j \in [4p - 4]$. We then add a $(2\lambda - 1)$-degree-gadget at w_t and a 4λ-degree-gadget on each $u_t^{(p-1)i-j}$ for any $i \in [4]$ and $j \in [\lambda - 1]$. For $r \in [4]$ let $x_t^r = u_t^{r(p-1)-p+2}$ and $\bar{x}_t^r = u_t^{r(p-1)-p+3}$. Set $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ and $\overline{X}_t = \{\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \bar{x}_t^4\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated* (resp. *negated) literal vertices* of H_t. See Figure 4.5.

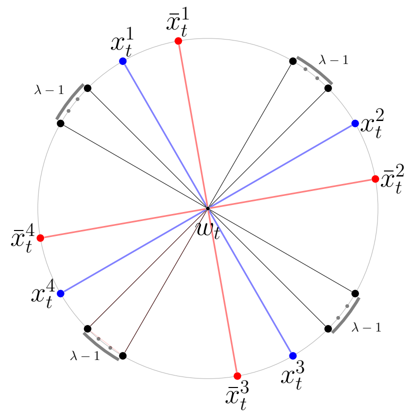


Figure 4.5: **The variable gadget H_t.** The center vertex w_t is in a $(2\lambda-1)$-degree-gadget. There are four sets of $\lambda - 1$ vertices (in black) adjacent to the center vertex w_t; each of them is in a 4λ-degree-gadget. Blue and red vertices are respectively non-negated and negated literal vertices. All blue ones (resp. red ones) are adjacent to w_t if and only if $\phi(x_t) = true$ (resp. $false$).

- For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$ we identify y_1, y_2, y_3 into a *clause vertex* c_j, where $y_i = x_t^r$ if $\ell_i = x_t$ and ℓ_i is the r-th occurrence of x_t, and $y_i = \bar{x}_t^r$ if $\ell_i = \bar{x}_t$ and is the r-th occurrence of x_t.

We will prove that there exists an assignment ϕ satisfying Φ if and only if there is an $(F, H, 6\lambda - 1)$-graph G. The general idea is that a variable $x_t = true$ (resp. $false$) if and only if the vertices of X_t (resp. $\overline{X}_t$) have degree $2\lambda - 1$ in $G[V(H_t)]$ and so they are adjacent to the center vertex while those one of the other set are not.

Assume that there exists an assignment ϕ satisfying Φ. Let G be the graph obtained as follows.

For each $t \in [n]$, let $(v_0, v_1, \ldots, v_{p-1})$ be an ordering of $V(F)$ such that $N_F(v_0) = \{v_{p-\lambda+1}, \ldots, v_{p-1}\} \cup \{v_1\}$ if $\phi(x_t) = true$ and $N_F(v_0) = \{v_{p-\lambda+1}, \ldots, v_{p-1}\} \cup \{v_2\}$ if $\phi(x_t) = false$. For every $j \in [4p-4]$, we let $G[S_t^j]$ be the copy of F in which w_t corresponds to v_0 and u_t^i for $i \in \{j \ldots, j+p-1\}$ corresponds to the vertex $v_{i'}$ such that $i \equiv i' \mod (p-1)$. Observe that each u_t^i corresponds to the same vertex of F in all the $p-1$ copies of F induced by the S_t^j to which it belongs. Therefore either u_t^i is not adjacent to w_t and it has 2λ neighbors in $G[H_t]$ or u_t^i is adjacent to w_t and it has $2\lambda - 1$ neighbors in $G[H_t]$. In particular, if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$), then all vertices of X_t (resp. $\overline{X}_t$) have degree $2\lambda - 1$ in $G[H_t]$ In addition, for every d-degree-gadget D at some vertex v, we let $G[V(D)]$ be an $(F, D, 6\lambda - 1)$-graph in which v has degree d.

Let us check that every vertex has degree at most $6\lambda - 1$ in G.

- Each center vertex w_t has degree $2\lambda - 1$ in its $(2\lambda - 1)$-degree-gadget and it is adjacent to 4λ vertices in H_t, so $6\lambda - 1$ in total.
- Every vertex in $\{u_t^{(p-1)i-j} \mid i \in [4] \text{ and } j \in [\lambda - 1]\}$ has $2\lambda - 1$ neighbors in H_t and 4λ other in its 4λ-degree-gadget. Hence its total degree is $6\lambda - 1$.
- Every vertex in $U_t \setminus \{u_t^{(p-1)i-j} \mid i \in [4] \text{ and } j \in [\lambda - 1]\}$ which is not identified in a clause vertex has only neighbors in H_t and thus degree at most $2\lambda < 6\lambda - 1$.
- Each clause vertex is the identification of three literal vertices which have degree 2λ or $2\lambda - 1$ in their variable gadgets. Moreover, at least one of the literals is true, so at least one of those vertices has only $2\lambda - 1$ neighbors in its variable gadget. Hence its degree in G is at most $6\lambda - 1$.

Hence, G is an $(F, H, 6\lambda - 1)$-graph.

Conversely, assume that G is an $(F, H, 6\lambda - 1)$-graph.
Consider a variable gadget H_t. The center vertex w_t has degree at least $2\lambda - 1$ in its $(2\lambda - 1)$-degree-gadget, so it has at most 4λ neighbors in H_t. Since w_t has degree at least λ in each of the S_t^j; and hyperedges S_t^j, S_t^{j+p-1}, S_t^{j+2p-2}, S_t^{j+3p-3} pairwise intersect only in w_t. So this vertex has exactly λ neighbors in each of these sets. Hence w_t has exactly λ neighbors in each S_t^j. Furthermore, if u_t^j is adjacent to w_t, then w_t has $\lambda - 1$ neighbors in $\{u_t^{j+1}, \ldots, u_t^{j+p-2}\}$ and so u_t^{j+p-1} is adjacent to w_t because S_t^{j+1} contains λ neighbors of w_t. In particular, the vertices of X_t (resp. $\overline{X}_t$) are either all adjacent to w_t or all non-adjacent to w_t.

Now each of the $\lambda - 1$ vertices in $\{u_t^{p-r+2}, \ldots, u_t^{p-1}\}$ is in a 4λ-gadget in which it has degree 4λ. Therefore, it has degree $2\lambda - 1$ in H_t and must be adjacent to w_t. Hence at most one vertex in $\{x_t^1, \bar{x}_1^t\}$ is adjacent to w_t. Thus the vertices of X_t and those of $\overline{X}_t$ cannot be simultaneously adjacent to w_t.

Let ϕ be the truth assignment defined by $\phi(x_t) = true$ if w_t is adjacent to X_t, and

$\phi(x_t) = false$ otherwise. In any clause vertex c_j, we identified three literal vertices corresponding to the three literals. But c_j has degree at most $6\lambda - 1$, so there is at least one literal vertex having degree $2\lambda - 1$ in its variable gadget. This implies that this literal is true. Therefore, ϕ satisfies Φ. $\square$

Complete graph minus an edge

Theorem 42. *$(\Delta \leq 3p-1)$-K_p^--*OVERLAY *is* NP*-complete for all $p \geq 3$.*

Proof. Reduction from (3,4)-SAT. Given a formula Φ of this problem, we build a hypergraph H as follows.

- For each variable x_t, we add a *variable gadget* H_t containing a *center set* C_t of size $p-2$, a set U_t of 8 vertices $U_t = \{u_t^1, \dots, u_t^8\}$, and 8 hyperedges $S_t^i = C_t \cup \{u_t^i, u_t^{i+1}\}$ (superscripts are modulo 8) for $i \in [8]$. Set $X_t = \{u_t^{2i-1} \mid i \in [4]\}$ and $\overline{X}_t = \{u_t^{2i} \mid i \in [4]\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated literal vertices* (resp. *negated literal vertices*).
- For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we add a *clause vertex* c_j in which, for each literal ℓ_i which is the r-th occurrence of the variable x_t, we identify u_t^{2r-1} (resp. u_t^{2r}) if $\ell_i = x_t$ (resp. $\ell_i = \bar{x}_t$).
- In any center set C_t, if $p = 3$ which $|C_t| = 1$, we add a $(2p-2)$-degree-gadget at this vertex; if $p \geq 4$, we add a $(2p-4)$-degree-gadget at $\max\{0, 6-p\}$ vertices and $(2p-5)$-degree-gadget at $\min\{4, 2p-8\}$ vertices among the other ones.

We will show that there is an assignment ϕ satisfying Φ if and only there is a $(K_p^-, H, 3p-1)$-graph G.

Assume that ϕ satisfies Φ, then we construct G as follows.

In a variable gadget H_t, for every $i \in [8]$, we let $G[S_t^i]$ be a copy of K_p^- in which $G[H_t]$ is such that

- every vertex in C_t which is not in any degree-gadget is a $(p-1)$-vertex, so it is adjacent to all vertices in H_t;
- if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$), then each vertex in X_t (resp. $\overline{X}_t$) is a $(p-2)$-vertex in every hyperedge containing it and each vertex in $\overline{X}_t$ (resp. X_t) is a $(p-1)$-vertex in every hyperedge containing it.
- any vertex in C_t which is in a d-degree-gadget is adjacent to all vertices in H_t except $p + 5 - (3p - 1 - d)$ literal vertices in exactly one of the two sets $X_t, \overline{X}_t$.

For any d-degree-gadget D at a vertex v, let $G[V(D)]$ be a $(K_p^-, D, 3p-1)$-graph in which v has degree d.

Let us check that $\Delta(G) \leq 3p - 1$.

- Each vertex in C_t which is not in any degree-gadget is adjacent to all vertices in H_t. So it has degree at most $p + 5$ in G.
- Each vertex in C_t which is in a d-degree-gadget has degree d in its degree-gadget and is adjacent to $3p - 1 - d$ vertices in H_t. So it has degree $3p - 1$ in G.

- Any literal vertex which is not identified in any clause vertex has degree either $p-1$ or p in its variable gadget (adjacent to $p-3$ vertices of C_t and two literal vertices which each is in the same hyperedge). Thus, it has degree less than $3p-1$.
- Each clause vertex is the identification of three literal vertices. Each of those has degree either $p-1$ or p in its variable gadget. Moreover at least one of the literals is true so its corresponding literal vertex has degree $p-1$ in its variable gadget. Therefore the clause vertex has degree at most $3p-1$.
- Any vertex which is in a degree-gadget but in no variable gadget belongs to at most two hyperedges, then it has degree at most $2(p-1) < 3p-1$.

Hence, G is a $(K_p^-, H, 3p-1)$-graph.

Conversely, assume that G is a $(K_p^-, H, 3p-1)$-graph. For every hyperedge S of H, let F_S be a subgraph of $G[S]$ isomorphic to K_p^-.

Claim 42.1. *Any $G[H_t]$ satisfies the following.*

(i) *$G[C_t]$ is complete.*

(ii) *For $i \in [8]$, $u_t^i u_t^{i+1} \in E(G[H_t])$ (superscripts are modulo 8).*

(iii) *All vertices of exactly one of $X_t, \overline{X}_t$ are $(p-2)$-vertices in all hyperedges of H_t which it belongs and those of the other are $(p-1)$-vertices.*

Proof of claim: (i) Assume that $G[C_t]$ is not complete, then there is an edge $uv \notin G[C_t]$ for $u, v \in V(C_t)$. Since all hyperedges of H_t is F-overlaying and F is a clique minus an edge, so the edge uv is the removing edge from the clique K_p in all hyperedges of H_t. It implies that in any hyperedge containing a literal vertex, this one must be adjacent to all vertices. Hence every literal vertex has degree p in $G[H_t]$ and thus any clause vertex must have degree $3p > 3p-1$ which is a contradiction.

(ii) Assume for a contradiction that there is $i \in [8]$ such that $u_t^i u_t^{i+1}$ is not an edge of $G[H_t]$. Thus, with the six other literal vertices, there are at most three non-edges between them to vertices in C_t (because otherwise, there are two non-edges in an induced subgraph of $G[H_t]$ by a hyperedge). However, since any vertex in C_t which is in a d-degree-gadget has degree at least d in its degree-gadget, at most $3p-1-d$ in $G[H_t]$. Hence, they must be non-adjacent to at least 4 literal vertices which is a contradiction.

(iii) By (ii), there are at least four none-edges and they are not in $G[C_t]$ by (i), then each of them is between a literal vertex and a vertex in C_t which belongs to a degree-gadget (there are at most 4 such vertices). Moreover, since any hyperedge S_t^i for $i \in [8]$ is F-overlaid, then there is at most one non-edge in $G[S_t^i]$. Assume $wu_t^i \notin E(G[H_t])$ for some $w \in C_t$ and it belongs to a degree-gadget, so u_t^i is $p-2$ in hyperedges which contain it. Thus, there are four none-edges and all vertices of exactly one of the sets $X_t, \overline{X}_t$ are $(p-2)$-vertices in hyperedges it belong and those of the other set are $(p-1)$-vertices. $\triangleleft$

By Claim 42.1, we define an assignment ϕ by $\phi(x_t) = true$ (resp. $\phi(x_t) = false$) if all vertices in X_t are $(p-2)$-vertices (resp. $(p-1)$-vertices) in the hyperedges of H_t to which they belong.
Observe that, each of these $(p-2)$-vertices has degree $p-1$ (it is adjacent to $p-3$ vertices of C_t and two literal vertices) and each of those $(p-1)$-vertices has degree p (it is adjacent to all vertices of C_t and two literal vertices) in $G[H_t]$.

A clause vertex c_j is the identification of three literal vertices. Since it has degree at most $3p-1$, then at least one of those literal vertices has degree at most $p-1$ in its variable gadget. Hence, this vertex is a $(p-2)$-vertex in the hyperedges of H_t to which it belongs. Thus this vertex corresponds to a true literal in the clause C_j. Therefore, ϕ satisfies Φ. $\square$

Disjoint union of the complete bipartite graph $K_{a,a+1}$

In this section, we study on the family of disjoint union of the graph $K_{a,a+1}$. We aim to prove the following.

Theorem 43. *Let $rK_{a,a+1}$ be the disjoint union of r copies of $K_{a,a+1}$. Then $\mathrm{np}(rK_{a,a+1}) \leq 3a+5$.*

In order prove this theorem, we first prove Theorem 44 which show that $\mathrm{np}(K_{a,a+1}) \leq 3a+5$, and then deduce using Lemma 45.

Theorem 44. *$(\Delta \leq 3a+5)$-$K_{a,a+1}$-OVERLAY is NP-complete.*

Proof. Reduction from (3,4)-SAT. Given a formula Φ of (3,4)-SAT with variables $x_t, t \in [n]$ and clauses $C_j, j \in [m]$, we build a hypergraph H as follows.

- For each variable x_t, we add a *variable gadget* H_t containing a set C_t^1 of size a, a set C_t^2 of size $a-1$ and a set U_t of eight vertices $U_t = \{u_t^1, \dots, u_t^8\}$, and eight hyperedges $S_t^i = C_t^1 \cup C_t^2 \cup \{u_t^i, u_t^{i+1}\}$ (superscripts are modulo 8) for $i \in [8]$. Set $X_t = \{u_t^1, u_t^3, u_t^5, u_t^7\}$ and $\overline{X}_t = \{u_t^2, u_t^4, u_t^6, u_t^8\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated literal vertices* (resp. *negated literal vertices*).
- For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we add a *clause vertex* c_j in which, for each literal ℓ_i which is the rth occurrence of the variable x_t, we identify u_t^{2r-1} (resp. u_t^{2r}) if $\ell_i = x_t$ (resp. $\ell_i = \bar{x}_t$).
- We add degree-gadgets on some vertices:
  - we add a $(2a+2)$-degree-gadget at each of vertices in C_t^1.
  - we add a $(2a+1)$-degree-gadget at each of vertices in C_t^2.

We will show that there is an assignment ϕ satisfying Φ if and only there is a $(K_{a,a+1}, H, 3a+5)$-graph G.

Assume that ϕ satisfies Φ, then we construct G as follows.
In a variable gadget H_t, for every $i \in [8]$, we let $G[S_t^i]$ be a copy of $K_{a,a+1}$ such that

- every vertex in C_t^1 is an a-vertex and each vertex in C_t^2 is an $(a+1)$-vertex (so $G[C_t]$ is $K_{a,a-1}$ with partition (C_t^1, C_t^2));
- if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$, then each vertex in X_t (resp. $\overline{X}_t$) is an a-vertex in every hyperedge containing it and each vertex in $\overline{X}_t$ (resp. X_t) is an $(a+1)$-vertex in every hyperedge containing it.

For any d-degree-gadget D at a vertex v, let $G[D]$ be a $(K_{a,a+1}, D, 3a+5)$-graph in which v has degree d.

Let us check that $\Delta(G) \leq 3a+5$.

- Each vertex in C_t^1 has degree $2a+2$ in its $(2a+2)$-degree-gadget. It is also adjacent the $a-1$ vertices in C_t^2, and to the four vertices of exactly one of the two sets $X_t, \overline{X}_t$. Thus, this vertex has degree $3a+5$ in G.
- Each vertex in C_t^2 has degree $2a+1$ in its $(2a+1)$-degree-gadget. It is also adjacent the a vertices in C_t^1 and to the four vertices in exactly one of sets $X_t, \overline{X}_t$. Hence, it has degree $3a+5$.
- Any literal vertex which is not identified in any clause vertex has degree at most $a+2$ in its variable gadget. So, it has degree $a+2 < 3a+5$ in G.
- Each clause vertex is the identification of three literal vertices. Each of those has degree either $a+1$ or $a+2$ in its variable gadget. Moreover at least one of the literals is true so its corresponding literal vertex has degree $a+1$ in its variable gadget. Therefore the clause vertex has degree at most $2(a+2)+a+1 = 3a+5$.
- Any vertex which is in a degree-gadget but in no variable gadget has degree at most $2(a+1) < 3a+5$ since it belongs to at most two hyperedges.

Hence, G is a $(K_{a,a+1}, H, 3a+5)$-graph.

Conversely, assume that G is a $(K_{a,a+1}, H, 3a+5)$-graph. For every hyperedge S of H, let F_S be a subgraph of $G[S]$ isomorphic to $K_{a,a+1}$. Free to remove some edges, we may assume that G is the union of the F_S over all hyperedges S of H. We have the following.

Claim 44.1. *For every $t \in [n]$, the following hold.*

(i) *In a hyperedge of H_t, the two literal vertices cannot be both a-vertices or both $(a+1)$-vertices.*
(ii) *In every hyperedge of H_t, the vertices in C_t^1 are a-vertices and the vertices in C_t^2 are $(a+1)$-vertices.*
(iii) *The vertices of one of the two sets $X_t, \overline{X}_t$ are a-vertices in all hyperedges of H_t to which they belong, and the vertices of the other of those sets are $(a+1)$-vertices in all hyperedges of H_t.*

Proof of claim: Observe that any vertex in C_t^1 is in a $(2a+2)$-degree-gadget, so it has degree at most $a+3$ in $G[H_t]$. Similarly, any vertex in C_t^2 is in a $(2a+1)$-degree-gadget, so it has degree at most $a+4$ in $G[H_t]$.

(i) Assume for a contradiction that there is $i \in [8]$ such that u_t^i, u_t^{i+1} are both a-vertices in S_t^i. There are $a-1$ other a-vertices in S_t^i. Thus, at least one vertex v in C_t^1 is an $(a+1)$-vertex in S_t^i, and thus adjacent to u_t^i, u_t^{i+1} and the $a-1$ other a-vertices in S_t^i which are in $C_t^1 \cup C_t^2$.

Assume for a contradiction that v is adjacent to exactly $a-1$ vertices in $C_t^1 \cup C_t^2$. Then because v has degree at least a in every hyperedge, it must be adjacent to at least one literal vertex in each $S_t^{i'}$ for all $i' \in [8]$. In particular v is adjacent to at least one literal vertex in S_t^{i+2}, S_t^{i+4}, and S_t^{i+6}. Hence v has degree $a+4$ in $G[H_t]$, a contradiction to the above observation.

Consequently, v is adjacent to at least a and at most $a+1$ vertices in $C_t^1 \cup C_t^2$.

- If v is adjacent to exactly a vertices in $C_t^1 \cup C_t^2$, then there is a vertex u in $C_t^1 \setminus \{v\}$ which is adjacent to v since there are only $a-1$ vertices in C_t^2. Vertex u has degree at least a in S_t^i. Since v has degree $a+2$ in $S_t^i \cup C_t^1 \cup C_t^2$, it is adjacent to at most one vertex, among the six literal vertices u_t^{i+1+j}, $j \in [6]$. Hence there are two hyperedges S, S' in $\{S_t^{i+2}, S_t^{i+4}, S_t^{i+6}\}$ such that v is adjacent to no literal vertex of S and S'. Now in each of those two hyperedges, v has degree exactly a. Hence it must be an a-vertex, and each of its neighbors, including u, is an $(a+1)$-vertex and thus is adjacent to the two literal vertices. Hence u is adjacent to at least $a+4$ vertices in $G[H_t]$ (at least a in S_t^i plus the four literal vertices of S and S'). This is a contradiction.
- If v is adjacent to $a+1$ vertices in $C_t^1 \cup C_t^2$, then there are two vertices u, u' in $C_t^1 \setminus \{v\}$ which are adjacent to v and each of them has degree at least a in S_t^i. Since v has degree $a+2$ in $S_t^i \cup C_t^1 \cup C_t^2$, it is not adjacent to any of the six other literal vertices than u_t^i, u_t^{i+1}. Consider the three hyperedges $S_t^{i+2}, S_t^{i+4}, S_t^{i+6}$;
  - if v is an $(a+1)$-vertex in one of these hyperedges, then u, u' must be a-vertices and thus adjacent to the two literal vertices in this hyperedge.
  - if v is an a-vertex in one of those hyperedges, then at least one of u, u' is an $(a+1)$-vertex in this hyperedge and is adjacent to its two literal vertex.

  Thus at least one of u,u' is adjacent to at least four literal vertices in $S_t^{i+2} \cup S_t^{i+4} \cup S_t^{i+6}$, and so has degree at least $a+4$, a contradiction.

This proves that the two literal vertices of a hyperedge of H_t are not both a-vertices.

Let us now prove that the two literal vertices of a hyperedge of H_t cannot be both $(a+1)$-vertices. Assume for a contradiction that there is $i \in [8]$ such that u_t^i, u_t^{i+1} are both $(a+1)$-vertices. Any a-vertex x in S_t^i is adjacent to u_t^i, u_t^{i+1} and at least $a-2$ vertices in $C_t^1 \cup C_t^2$. If x is adjacent to exactly $a-2$ (resp. $a-1$) vertices in $C_t^1 \cup C_t^2$, then, since it has degree at least a in any hyperedge, it is adjacent to all six (resp. at

least three) literal vertices in $S_t^{i+2} \cup S_t^{i+4} \cup S_t^{i+6}$. Thus x has degree $a+6$ (resp. $a+4$) in $G[H_t]$, a contradiction. Hence every a-vertex in S_t^i has at least a neighbors in $C_t^1 \cup C_t^2$. There are $a+1$ a-vertices in S_t^i, so there must be one, say v, in C_t^1. It has degree at most $a+3$ in $G[H_t]$ and at least $a+2$ in S_t^i. Thus it is adjacent to at most one literal vertex in $S_t^{i+2} \cup S_t^{i+4} \cup S_t^{i+6}$. Hence v is not adjacent to the literal vertices of two hyperedges S, S' in $S_t^{i+2} \cup S_t^{i+4} \cup S_t^{i+6}$. Thus the literal vertices of the hyperedge S (resp. S') are both in a same part of F_S (resp. $F_{S'}$), and so they are $(a+1)$-vertices.
Now in each hyperedge of H_t, there are more a-vertices than $(a+1)$-vertices. Thus there is a vertex z which is an a-vertex in at least three hyperedges S_1, S_2, S_3 in $S_t^i \cup S_t^{i+2} \cup S_t^{i+4} \cup S_t^{i+6}$. In any of these hyperedges at least one of the literal vertices is an $(a+1)$-vertex, and in at least two of them the two literal vertices are $(a+1)$-vertices. Hence z is adjacent to at least five literal vertices. Moreover, as above, we can show that z has at least a neighours in $C_t^1 \cup C_t^2$. Thus z has degree at least $a+5$ in $G[H_t]$, a contradiction. This completes the proof of (i).

(ii) Assume for a contradiction that a vertex $w \in C_t^1$ is an $(a+1)$-vertex in S_t^i. By (i), w is adjacent to a literal vertex in S_t^i, and so it is adjacent to a other vertices in $C_t^1 \cup C_t^2$. Furthermore, by (i), in each hyperedge of H_t, w is adjacent to a literal vertex (either to an a-vertex or an $(a+1)$-vertex). Thus w is adjacent to four literal vertices in H_t, and so has degree at least $a+4$ in $G[H_t]$, a contradiction. Therefore the a vertices of C_t^1 are a-vertices. Moreover, by (i), one of the literal vertex of each S_t^i is an a-vertex. Therefore all vertices of C_t^2 must be $(a+1)$-vertices.

(iii) Let v be a vertex in C_t^1. It is an a-vertex in each S_t^i, so by (i) it is adjacent to one vertex in $\{u_t^i, u_t^{i+1}\}$ for all $i \in [8]$ and it is adjacent to the $a-1$ vertices of C_t^2. But v has degree at most $a+3$ in $G[H_t]$, so v is either adjacent to all vertices of X_t and non-adjacent to all vertices of $\overline{X}_t$, or non-adjacent to all vertices of X_t and adjacent to all vertices of $\overline{X}_t$. $\triangleleft$

By Claim 44.1, we define a truth assignment ϕ by $\phi(x_t) = true$ (resp. $\phi(x_t) = false$) if all vertices in X_t are a-vertices (resp. $(a+1)$-vertices) in the hyperedges of H_t to which they belong.

Observe that, by Claim 44.1, if a literal vertex u_t^i is an $(a+1)$-vertex in the hyperedges of H_t to which it belongs then it has degree at least $a+2$ in $G[H_t]$ because it is adjacent to the a vertices of C_t^1 and the two literal vertices u_t^{i-1}, u_t^{i+1}.

A clause vertex c_j is the identification of three literal vertices. Since it has degree at most $3a+5$, then at least one of those literal vertices has degree at most $a+1$ in its variable gadget. By the above observation, this vertex is an a-vertex in the hyperedges of H_t to which it belongs. Thus this vertex corresponds to a true literal in the clause C_j. Therefore, ϕ satisfies Φ. $\square$

Lemma 45. *Let r be a positive integer. If $(\Delta \leq k)$-$K_{a,a+1}$-*OVERLAY *is* NP*-complete, then $(\Delta \leq k)$-$rK_{a,a+1}$-*OVERLAY *is* NP*-complete.*

Proof. $K_{a,a+1}$ has $a+1$ vertices of degree a and a vertices of degree $a+1$. Hence, in $rK_{a,a+1}$, there are $r(a+1)$ vertices of degree a and ra vertices of degree $a+1$.

We shall give a reduction from $(\Delta \leq k)$-$K_{a,a+1}$-OVERLAY to $(\Delta \leq k)$-$rK_{a,a+1}$-OVERLAY.
Let H be a $(2a+1)$-uniform hypergraph. We construct an $r(2a+1)$-uniform hypergraph H' from H as follows. We create a set A of $(r-1)(a+1)$ vertices, a set B of $(r-1)a$ vertices, and a set C of $2a+1$ vertices. We add the hyperedge $S_C = A \cup B \cup C$ to $E(H')$, and for every hyperedge S of H, we add the hyperedge $S' = S \cup A \cup B$ to $E(H')$. Finally, we add a $(k-a)$-degree-gadget at every vertex in A and a $(k-a-1)$-degree-gadget at every vertex in B.

Let us prove that there is a $(K_{a,a+1}, H, k)$-graph G if and only if there is an $(rK_{a,a+1}, H', k)$-graph G'.

Assume that G is a $(K_{a,a+1}, H, k)$-graph. We construct G' from G as follows. Let $G'[H] = G[H]$, so $G'[S] = G[S]$ for each $S \in E(H)$; let $G'[C]$ be a copy of $K_{a,a+1}$; let $G'[A \cup B]$ be a copy of $(r-1)K_{a,a+1}$ in which every vertex in A has degree a and every vertex in B has degree $a+1$; for each d-degree-gadget D at a vertex v, let $G'[D]$ be an $(rK_{a,a+1}, D, k)$-graph in which v has degree d. Clearly, for any $S' \in E(H')$, $G'[S']$ contains $rK_{a,a+1}$ and so does $G'[S_C]$. Moreover, one easily checks that every vertex of G' has degree at most k. Therefore, G' is an $(rK_{a,a+1}, H', k)$-graph.

Assume now that there is an $(rK_{a,a+1}, H', k)$-graph G'. Every vertex $v \in A$ is in a $(k-a)$-degree-gadget, so it has degree at most a in $G'[V(H) \cup A \cup B \cup C]$. Thus it must be an a-vertex in every hyperedge S' for all $S \in E(H)$.

Let v be a vertex in A. It is adjacent to $a-i$ vertices in B. Then v must be adjacent to at least i vertices in C and i vertices in $V(H)$. Thus the degree of v is at least $a+i$ in $G'[V(H) \cup A \cup B \cup C]$. Therefore $i = 0$, so v is adjacent to a vertices in B and no vertex in $V(H) \cup C$.

This implies that there are $(d-1)a(a+1)$ edges between A and B. But every vertex $u \in B$ is in a $(k-a-1)$-degree-gadget, and so has degree at most $a+1$ in $G'[V(H) \cup A \cup B \cup C]$. Thus, each vertex in B has $a+1$ neighbors in A, and is adjacent to vertex in $V(H) \cup C$.

Consider now a hyperedge $S' = S \cup A \cup B$. The graph $G'[S']$ contains $rK_{a,a+1}$. Since there is no edge between $A \cup B$ and $V(H)$, necessarily $G'[S]$ contains $K_{a,a+1}$. So S is $K_{a,a+1}$-overlaid by G'. Consequently, $G = G'[V(H)]$ is a $(K_{a,a+1}, H, k)$-graph. $\square$

4.3 Proof of Theorem 34

The aim of this section is to prove Theorem 34. The proof divides into four cases, Theorem 41, Theorem 46, Theorem 47 and Theorem 48 as follows.

Proof of Theorem 34 (assuming Theorems 46, 47 and 48). Let F be a graph with degree values $1 \leq \delta(F) = \lambda_1 < \cdots < \lambda_t = \Delta(F)$.

If $t = 1$, (i.e. F is regular), then we have the result by Theorem 41. Henceforth, we may assume that $t \geq 2$.

If there exists $i \in [t-1]$ such that $\lambda_{i+1} > \lambda_i + 1$, then Theorem 46 yields the result. Henceforth, we may assume that $\lambda_{i+1} = \lambda_i + 1$ for all $i \in [t-1]$.

If $t \geq 3$, then $\lambda_t + \lambda_1 \geq 2\lambda_2$ and Theorem 47 yields the result. Henceforth, we may assume that $t = 2$ which we then have the result by Theorem 48. $\square$

It thus remains to prove Theorems 46, 47 and 48.

The proofs of the first two are reductions from 3-COLORING on 4-regular graphs which are similar to the one used to prove Theorem 37. Given a 4-regular graph G, we build a hypergraph H which includes, for each vertex $v \in V(G)$, a *vertex gadget* with three hyperedges which makes three choices of degrees on vertices c_v^1, c_v^2, c_v^3 (as three colors labeled 1, 2, 3 of vertex v) and a *color gadget* represented as a binary tree with 4 leaves which copies each choice to four (leaves) vertices in other hyperedges (with respect to four neighbors of $v \in V(G)$). For any edge uv, we simply identifies the two leaves of u, v. The idea is that for a proper coloring c of G, $c(v)$ corresponds to a vertex c_v^i having a certain degree d; then $c(v) = i$ if and only if c_v^i as degree d in its vertex gadget (see Figure 4.6). However, the set of hyperedges which are in a color gadget of the two theorems are different, see Figure 4.7 in Theorem 46 and Figure 4.8 in Theorem 47.

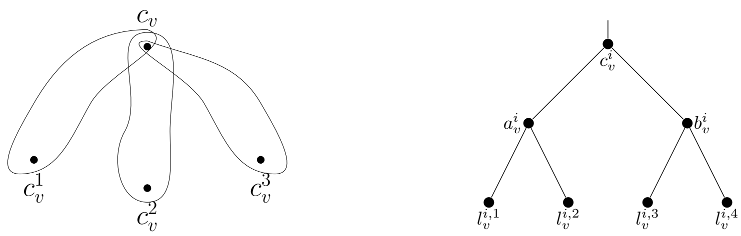


Figure 4.6: **The construction of the reduction.** The vertex gadget for vertex v (left) and the color gadget represented as a binary tree with 4 leaves (right). In the construction, each edge of this tree is replaced by hyperedges such that the degree of the root c_v^i is transmitted to all its descendants.

Theorem 46. *Let F be a graph on p vertices with degree values $1 \leq \lambda_1 < \cdots < \lambda_t$. If there exists $i^* \in \{2, \ldots, t\}$ such that $\lambda_{i^*} \geq \lambda_{i^*-1} + 2$, then there is k such that $(\Delta \leq k)$-F-*OVERLAY *is* NP-*complete.*

Proof. Set $k = \max\{2\lambda_t, 2\lambda_{i^*} + \lambda_{i^*-1} + \lambda_1\}$. We give a reduction from 3-COLORING on 4-regular graphs.

Given a 4-regular graph G, we build a hypergraph H as follows.

- For each vertex $v \in V(G)$, we create a *vertex gadget* H_v with three hyperedges $S_v^i = \{c_v, c_v^i\} \cup X_v^i \cup Y_v^i$ for $i \in [3]$ where $|X_v^i| = \sum_{j=1}^{i^*-1} \alpha_j - 1$, $|Y_v^i| = p - |X_v^i| - 2$. We add a $(k - \lambda_{i^*} + 1)$-degree-gadget at each vertex $x \in X_v^i$ for $i \in [3]$, a $(k - 2\lambda_{i^*} - \lambda_{i^*-1})$-degree-gadget at c_v. We say that S_v^i is the *parent hyperedge* of c_v^i for each $i \in [3]$.
- For each vertex v and each $i \in [3]$, we construct a *color gadget* H_v^i for $i \in [3]$ as follows.
  - We create a binary tree T_v^i with vertex set $\{c_v^i, a_v^i, b_v^i, \ell_v^{i,1}, \ell_v^{i,2}, \ell_v^{i,3}, \ell_v^{i,4}\}$ and edge set $\{c_v^i a_v^i, c_v^i b_v^i, a_v^i \ell_v^{i,1}, a_v^i \ell_v^{i,2}, b_v^i \ell_v^{i,3}, b_v^i \ell_v^{i,4}\}$, rooted at c_v^i. In this tree, a_v^i and b_v^i are the children of c_v^i, $\ell_v^{i,1}$ and $\ell_v^{i,2}$ are the children of a_v^i, and $\ell_v^{i,3}$ and $\ell_v^{i,4}$ are the children of b_v^i.
  - For any vertex $x \in \{c_v^i, a_v^i, b_v^i\}$, let y_1, y_2 be its children in T_v^i, and let $e_1 = xy_1, e_2 = xy_2$. We first add a $(k - 2\lambda_{i^*} + 1)$-degree-gadget at x. Then we construct an x-edge-gadget as follows: we add a set A_x of $\sum_{j=1}^{i^*-1} \alpha_j - 1$ new vertices and a set B_x of $p - |A_x| - 2$ new vertices, the hyperedges $S(e_1) = \{x, y_1\} \cup A_x \cup B_x$ and $S(e_2) = \{x, y_2\} \cup A_x \cup B_x$, and a $(k - \lambda_{i^*} + 1)$-degree-gadget at every vertex $a \in A_x$. For convenience, we say that $S(xy_1)$ (resp. $S(xy_2)$) is the parent hyperedge of y_1 (resp. y_2). Moreover, for any leaf $\ell_v^{i,j}$, we denote by $S_v^{i,j}$ the hyperedge containing the vertex $\ell_v^{i,j}$.
- For every vertex $v \in V(G)$, let $e_v^1, e_v^2, e_v^3, e_v^4$, be an ordering of the edges incident to v. For each edge $uv \in E(G)$, let j_u and j_v be the indices such that $uv = e_u^{j_u} = e_v^{j_v}$. Then, for all $i \in [3]$, we identify the vertices ℓ_u^{i,j_u} and ℓ_v^{i,j_v} and we add a $(k - \lambda_{i^*} - \lambda_1)$-degree-vertex at this vertex.

Note that each of the d-degree-gadgets exists because we have $d \geq \lambda_1$ by our choice of k.

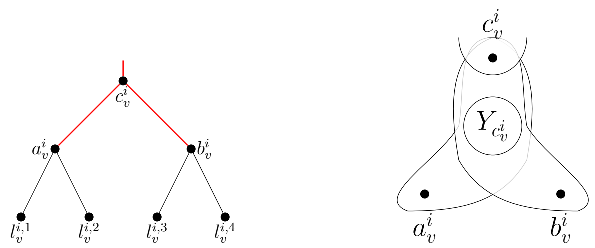


Figure 4.7: The binary tree T_v^i and an example of the edge-gadget at vertex c_v^i (corresponding to red edges).

Let us now prove that there is a proper 3-coloring of G if and only if there is an (F, H, k)-graph G^*.

Assume first that there is an (F, H, k)-graph G^*. For sake of clarity, we say that a vertex has degree d in a set S if it has degree d in $G^*[S]$.

Let $v \in V(G)$. The vertex c_v has degree at least $(k - 2\lambda_{i^*} - \lambda_{i^*-1})$ in its $(k - 2\lambda_{i^*} - \lambda_{i^*-1})$-degree-gadget. Hence c_v has degree at most $2\lambda_{i^*} + \lambda_{i^*-1}$ in $S_v^1 \cup S_v^2 \cup S_v^3$. But those hyperedges pairwise intersect in $\{c_v\}$. Thus there is $i \in [3]$ such that c_v has degree less than λ_{i^*} in S_v^i. Moreover, since any vertex $x \in X_v^i$ has degree at least $k - \lambda_{i^*} + 1$ in its $(k - \lambda_{i^*} + 1)$-degree-gadget, so it has degree less than λ_{i^*} in S_v^i. Thus c_v^i must have degree at least λ_{i^*} in S_v^i. Therefore, we can define a 3-coloring ϕ by $\phi(v) = i$ where i is an index such that c_v^i has degree at least λ_{i^*} in S_v^i.
Let us now prove that ϕ is proper. We need the following claim.

Claim 46.1. *Let $v \in V(G)$ and $i \in [3]$. If c_v^i has degree at least λ_{i^*} in S_v^i, then so does the leaf $\ell_v^{i,j}$ in $S_v^{i,j}$ for all $j \in [4]$.*

Proof of claim: It suffices to prove that for any $x \in \{c_v^i, b_v^i, a_v^i\}$, if x has degree at least λ_{i^*} in its parent hyperedge, then both y_1, y_2 have degree at least λ_{i^*} in their parent hyperedges.

Assume that x has degree at least λ_{i^*} in its parent hyperedge. Since x has degree at least $k - 2\lambda_{i^*} + 1$ in its $(k - 2\lambda_{i^*} + 1)$-degree-gadget, it has degree at most $\lambda_{i^*} - 1$ in $S(xy_1) \cup S(xy_2)$. Moreover, any $a \in A_x$ has degree at least $k - \lambda_{i^*} + 1$ in its $(k - \lambda_{i^*} + 1)$-degree-gadget and so has degree less than λ_{i^*} in $S(xy_1) \cup S(xy_2)$ and so in each of $S(xy_1), S(xy_2)$. Since A_x is of size $\sum_{j=1}^{i^*-1} \alpha_1 - 1$, the vertex y_1 (resp. y_2) must have degree at least λ_{i^*} in $S(xy_1)$ (resp. $S(xy_2)$). $\triangleleft$

Consider an edge $uv \in E(G)$, $i \in [3]$. The vertex $\ell = \ell_u^{i,j_u} = \ell_v^{i,j_v}$ has degree at least $k - \lambda_{i^*} - \lambda_1$ in its $(k - \lambda_{i^*} - \lambda_1)$-degree-gadget and is the unique common vertex of the hyperedges S_u^{i,j_u} and S_v^{i,j_v}. Therefore it has degree λ_{i^*} in at most one of S_u^{i,j_u}, S_v^{i,j_v}. Hence, by the Claim 46.1, at most one of c_u^i, c_v^i has degree λ_{i^*} in its parent hyperedge. Thus at most one of u, v is colored i by ϕ. Therefore, ϕ is a proper 3-coloring of G.

Assume now that ϕ is a proper 3-coloring of G. We construct a graph G^* as follows.

- For any vertex gadget H_v, $i \in [3]$, let $G^*[S_v^i]$ be a copy of F in which every vertex in X_v^i has degree at most λ_{i^*-1}, every vertex in Y_v^i has degree at least λ_{i^*}, and c_v has degree λ_{i^*-1} (resp. λ_{i^*}) and c_v^i has degree λ_1 (resp. λ_{i^*}) in S_v^i if $\phi(v) = i$ (resp. $\phi(v) \neq i$).
- In every color gadget H_v^i, for $x \in \{c_v^i, b_v^i, a_v^i\}$ with children y_1 and y_2, let $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$) be two similar copies of F such that:
  - if $i \neq \phi(v)$, then x has degree λ_{i^*} in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ (and so at most $\lambda_{i^*} + 1$ in $G^*[S(xy_1) \cup S(xy_2)]$) and y_1 and y_2 have degree λ_1 in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ respectively.
  - if $i = \phi(v)$, then x has degree λ_{i^*-1} in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ (and so at most $\lambda_{i^*-1} + 1$ in $G^*[S(xy_1) \cup S(xy_2)]$) and y_1 and y_2 have degree λ_{i^*} in $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ respectively.

– every vertex in A_x has degree at most λ_{i^*-1} in both $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ and so at most $\lambda_{i^*}+1$ in $G^[S(xy_1) \cup S(xy_2)]$;
– every vertex in B_x is degree at least λ_{i^*} in both $G^*[S(xy_1)]$ and $G^*[S(xy_2)]$ and so at most $\lambda_t + 1$ in $G^*[S(xy_1) \cup S(xy_2)]$;

- For any d-degree-gadget D at vertex v, we let $G^*[V(D)]$ be an (F, D, k)-graph in which v has degree d.

By construction, G^* F-overlays H. Let us check that $\Delta(G^*) \leq k$. Let u be a vertex of G^*.

- If u is in at most two hyperedges (in particular, if = u is in Y_v^i or u is in B_x for x internal vertex in some T_v^i or u is only in a d-degree-gadget), then u has degree at most $2\lambda_t \leq k$.
- If $u \in X_v^i$ for $v \in V(G)$, then u has degree $k - \lambda_{i^*} + 1$ in its $(k - \lambda_{i^*} + 1)$-degree-gadget and at most λ_{i^*-1} in $S_v^i]$, thus u has degree at most $k - \lambda_{i^*} + \lambda_{i^*-1} + 1 \leq k$.
- If $u \in A_x$ for $v \in V(G)$ and x internal vertex in some tree T_v^i, then u has degree $k - \lambda_{i^*} + 1$ in its $(k - \lambda_{i^*} + 1)$-degree-gadget and at most $\lambda_{i^*-1} + 1$ in $G^*[S(xy_1) \cup S(xy_1)]$, thus u has degree at most $k - \lambda_{i^*} + \lambda_{i^*-1} + 2 \leq k$.
- For $u \in \{c_v^i, a_v^i, b_v^i\}$ for $i \in [3]$ with u parent of y_1, y_2, it has degree $k - 2\lambda_{i^*} + 1$ in its $(k - 2\lambda_{i^*} + 1)$-degree-gadget. And if $i = \phi(v)$ (resp. $i \neq \phi(v)$), then u has degree λ_{i^*} (resp. λ_1) in its parent hyperedge and $\lambda_{i^*-1} + 1$ (resp. $\lambda_{i^*} + 1$) in $G^*[S_{uy_1}^1 \cup S_{uy_2}^1]$. Hence u has degree at most $k - \lambda_{i^*} + \lambda_{i^*-1} + 2 \leq k$.
- Assume that u is the identification of ℓ_v^{i,j_v} and ℓ_w^{i,i_w} for an edge $vw \in E(G)$. First, u has degree $k - \lambda_{i^*} - \lambda_1$ in its $(k - \lambda_{i^*} - \lambda_1)$-degree-gadget. Moreover, since either $\phi(v) \neq i$ or $\phi(w) \neq i$, then u has degree λ_1 in one of S_v^{i,j_v}, S_w^{i,j_w} and at most λ_{i^*} in the other. Therefore, u has degree at most k in G^*.

Consequently, G^* is an (F, H, k)-graph. □

Theorem 47. *Let a graph F on p vertices with degree sequence $\mathbf{d} = (d_1, \ldots, d_p)$ such that $\lambda_t + \lambda_1 \geq 2\lambda_2$. Then there exists k such that $(\Delta \leq k)$-F-*Overlay *is* NP*-complete.*

Proof. Observe that the condition $\lambda_t + \lambda_1 \geq 2\lambda_2$ implies $t \geq 3$. Set $k = 2\lambda_t + \lambda_{t-1}$. We give a reduction from 3-Coloring on 4-regular graphs.

Given a 4-regular graph G, we build a p-uniform hypergraph H as follows.

- For each vertex $v \in V(G)$, we create a *vertex gadget* H_v with three hyperedges $S_v^i = \{c_v, c_v^i\} \cup X_v^i \cup Y_v^i$ for $i \in [3]$ where $|X_v^i| = \sum_{i=1}^{t-1} \alpha_i - 1$, $|Y_v^i| = p - \alpha_t - 1$. For $i \in [3]$, we add a $(k - \lambda_{t-1})$-degree-gadget at each vertex $x \in X_v^i$
We say that S_v^i is the parent hyperedge of each c_v^i, $i \in [3]$.
- For each vertex $v \in V(G)$ and each $i \in [3]$, we construct a *color gadget* H_v^i for $i \in [3]$ as follows.
  – We create a binary tree T_v^i with vertex set $\{c_v^i, a_v^i, b_v^i, \ell_v^{i,1}, \ell_v^{i,2}, \ell_v^{i,3}, \ell_v^{i,4}\}$ and edge set

$\{c_v^i a_v^i, c_v^i b_v^i, a_v^i \ell_v^{i,1}, a_v^i \ell_v^{i,2}, b_v^i \ell_v^{i,3}, b_v^i \ell_v^{i,4}\}$, rooted at c_v^i. In this tree, a_v^i and b_v^i are the children of c_v^i, $\ell_v^{i,1}$ and $\ell_v^{i,2}$ are the children of a_v^i, and $\ell_v^{i,3}$ and $\ell_v^{i,4}$ are the children of b_v^i.

– For each edge $e = xy$ of T_v^i with x the parent of y in T_v^i, we construct an *edge-gadget* containing x, y, a new vertex z_e, and four disjoint sets $U_e^1 W_e^1, U_e^2, W_e^2$ of new vertices, U_e^1 of size $\alpha_1 - 1$, W_e^1 of size $p - |U_e^1| - 1$, U_e^2 of size $p - \alpha_t - 1$, W_e^2 of size $\alpha_t - 1$. We add the hyperedges $S_e^1 = \{x, z_e\} \cup U_e^1 \cup W_e^1$ and $S_e^2 = \{z_e, y\} \cup U_e^2 \cup W_e^2$. We finally add a $(k - \lambda_t - 2\lambda_1)$-degree-gadget at x, a $(k - \lambda_1)$-degree-gadget at each vertex of U_e^1, a $(k - \lambda_t + 1)$-degree-gadget at each of U_e^2, and a $(k - \lambda_2 - \lambda_t + 1)$-degree-gadget pendant at z_e.

- For every vertex $v \in V(G)$, let $e_v^1, e_v^2, e_v^3, e_v^4$, be an ordering of the edges incident to v. For each edge $uv \in E(G)$, let j_u and j_v be the indices such that $uv = e_u^{j_u} = e_v^{j_v}$. Then, for all $i \in [3]$, we identify the vertices ℓ_u^{i,j_u} and ℓ_v^{i,j_v} and we add a $(k - 2\lambda_t + 1)$-degree-gadget at this vertex.

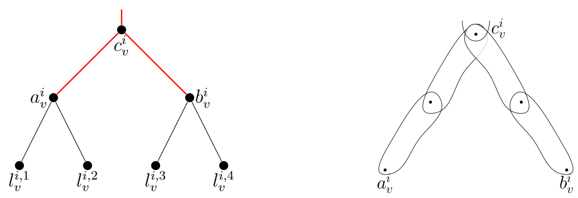


Figure 4.8: The binary tree T_v^i and an example of the edge-gadget at vertex c_v^i (corresponding to red edges).

Let us now prove that there is a proper 3-coloring of G if and only if there is an (F, H, k)-graph G^*.

Assume first that there is an (F, H, k)-graph G^*. For sake of clarity, we say that a vertex has degree d in a set S if it has degree d in $G^*[S]$.

Let $v \in V(G)$. The vertex c_v has degree at most $2\lambda_t + \lambda_{t-1}$ in $S_v^1 \cup S_v^2 \cup S_v^3$. But those hyperedges pairwise intersect in $\{c_v\}$. Thus there is $i \in [3]$ such that c_v has degree less than λ_t in S_v^i.

Moreover, each vertex $x \in X_v^i$ has degree at least $k - \lambda_{t-1}$ in its $(k - \lambda_{t-1})$-degree-gadget, and so at most λ_{t-1} in S_v^i. Together with c_v, there are $\sum_{i=1}^{t-1} \alpha_t$ vertices of degree at most λ_{t-1} in S_v^i. Thus c_v^i have degree λ_t in its parent hyperedge S_v^i. Therefore, we can define a 3-coloring ϕ by $\phi(v) = i$ where i is an index such that c_v^i has degree λ_t in S_v^i.

Let us now prove that ϕ is proper. We need the following claim.

Claim 47.1. *Let $v \in V(G)$ and $i \in [3]$. If c_v^i has degree λ_t in S_v^i, then so does any leaf $\ell_v^{i,j}$ in $S_v^{i,j}$ for $j \in [4]$.*

Proof of claim: It suffices to prove that for any $x \in \{c_v^i, b_v^i, a_v^i\}$, if x has degree λ_t in its parent hyperedge, then both y_1, y_2 have degree λ_t in their parent hyperedges.

Assume that x is a λ_t-vertex in its parent hyperedge. Since x has degree at least $k - \lambda_t - 2\lambda_1$ in its $(k - \lambda_t - 2\lambda_1)$-degree-gadget, and degree λ_t in its parent hyperedge, it has has degree at most $2\lambda_1$ in $S^1_{xy_1} \cup S^1_{xy_2}$, and so λ_1 in each of $S^1_{xy_1}, S^1_{xy_2}$. Let $e = xy$ be one of the two edges xy_1, xy_2. Any vertex in U_e^1 has degree at least $k - \lambda_1$ in its $(k - \lambda_1)$-degree-gadget, and thus λ_1 in S_e^1. It implies that z_e has degree at least λ_2 in S_e^1. Since it is also in a $(k - \lambda_2 - \lambda_t + 1)$-degree-gadget, z_e has degree less than λ_t in S_e^2. Moreover, any vertex in U_e^2 is in a $(k - \lambda_t + 1)$-degree-gadget, then none of them has degree λ_t in S_e^2 except those in W_e^2 which is of size $\alpha_t - 1$. Thus, y must have degree λ_t in S_e^2. $\triangleleft$

Consider an edge $uv \in E(G)$, $i \in [3]$. The vertex $\ell = \ell_u^{i,j_u} = \ell_v^{i,j_v}$ has degree at least $k - 2\lambda_t + 1$ in its $(k - 2\lambda_t + 1)$-degree-gadget and is the unique common vertex of the hyperedges S_u^{i,j_u} and S_v^{i,j_v}. Therefore it has degree λ_t in at most one of S_u^{i,j_u} and $G^* S_v^{i,j_v}$. Hence, by Claim 47.1, at most one of c_u^i, c_v^i has degree λ_t in its parent hyperedge. Thus at most one of u, v is colored i by ϕ. Therefore, ϕ is a proper 3-coloring of G.

Assume now that ϕ is a proper 3-coloring of G. We construct a graph G^* as follows.

- For any vertex gadget H_v, $i \in [3]$, let $G^*[S_v^i]$ be a copy of F in which every vertex in X_v^i has degree at most λ_{t-1}, every vertex in Y_v^i has degree λ_t, and c_v has degree λ_{t-1} (resp. λ_t) and c_v^i has degree λ_t (resp. λ_1) if $\phi(v) = i$ (resp. $\phi(v) \neq i$).
- In every color gadget H_v^i, for each edge $e = xy$ of T_v^i with x parent of y, let $G^*[S_e^1], G^*[S_e^2]$ be a copies of F such that:
  - every vertex in U_e^1 has degree λ_1;
  - every vertex in U_e^2 has degree at most λ_{t-1};
  - every vertex in W_e^2 has degree λ_t;
  - if $i = \phi(v)$, then x has degree λ_1 in S_e^1, z_e has degree λ_2 in S_e^1 and λ_{t-1} in S_e^2, and y has degree λ_t in S_e^2;
  - if $i \neq \phi(v)$, then x has degree λ_2 in S_e^1, z_e has degree λ_1 in S_e^1 and λ_t in S_e^2, and y has degree λ_1 in S_e^2..
- For any d-degree-gadget D at vertex v, we let $G^*[V(D)]$ be an (F, D, k)-graph in which v has degree d.

By construction, G^* F-overlays H. Let us check that $\Delta(G^*) \leq k$. Let u be a vertex of G^*.

- If u is in at most two hyperedges, (in particular if u is in Y_v^i, in $W_e^1 \cup W_e^2$ in an edge-gadget or only in a degree-gadget), then u has degree at most $2\lambda_t < k$ in G^*.

- If $u = c_v$, then it has degree λ_{t-1} in S_v^i for the index $i = \phi(v)$, and λ_t in S_v^i for the two indices $i \neq \phi(v)$. Hence c_v has degree $2\lambda_1 + \lambda_{t-1} = k$.
- If $u \in X_v^i$ for $v \in V(G)$, then u has degree $k - \lambda_{t-1}$ in its $(k - \lambda_{t-1})$-degree-gadget and at most λ_{t-1} in S_v^i, thus u has degree at most k in G^*.
- If $u \in U_e^1$ for some edge e of T_v^i, then u has degree $k - \lambda_1$ in its degree-gadget and λ_1 in S_e^1, thus u has degree k in G^*.
- If $u \in U_e^2$, then u has degree $k - \lambda_t + 1$ in its degree-gadget and at most λ_{t-1} in S_e^2, thus u has degree at most k in G^*.
- If $u \in \{c_v^i, a_v^i, b_v^i\}$ for $i \in [3]$ with children y_1, y_2, then u has degree $k - \lambda_t - 2\lambda_1$ in its $(k - \lambda_t - 2\lambda_1)$-degree-gadget. Moreover, if $i = \phi(v)$ (resp. $i \neq \phi(v)$), then u has degree λ_t (resp. λ_1) in its parent hyperedge and λ_1 (resp. λ_2) in both $S_{uy_1}^1, S_{uy_2}^1$. Hence u has degree at most $k - \lambda_t - 2\lambda_1 + \lambda_t + 2\lambda_1 = k$ (resp. $k - \lambda_t - 2\lambda_1 + \lambda_1 + 2\lambda_2 \leq k$ by the assumption $\lambda_t + \lambda_1 \geq 2\lambda_2$) in G^*.
- If $u = z_e$ for some edge e of T_v^i, then u has degree $k - \lambda_2 - \lambda_t + 1$ in its $(k - \lambda_2 - \lambda_t + 1)$-degree-gadget. Moreover, if $i = \phi(v)$ (resp. $i \neq \phi(v)$), then u has degree λ_2 (resp. λ_1) in S_e^1 and λ_{t-1} (resp. λ_t) in S_e^2. Hence, u has degree at most $k - \lambda_2 - \lambda_t + 1 + \lambda_2 + \lambda_{t-1} \leq k$ (resp. $k - \lambda_2 - \lambda_t + 1 + \lambda_1 + \lambda_t \leq k$) in G^*.
- Assume that u is the identification of ℓ_v^{i,j_v} and ℓ_w^{i,i_w} for an edge $vw \in E(G)$. First, u has degree $k - 2\lambda_t + 1$ in its $(k - 2\lambda_t + 1)$-degree-gadget. Moreover, since either $\phi(v) \neq i$ or $\phi(w) \neq i$, then u has degree less than λ_t in one of S_v^{i,j_v}, S_w^{i,j_w}. Therefore, u has degree at most k in G^*.

Consequently, G^* is an (F, H, k)-graph. □

Theorem 48. *Let F be a graph with α_1 vertices of positive degree λ_1 and $\alpha_2 = p - \alpha_1$ vertices of degree $\lambda_2 = \lambda_1 + 1$. Then $(\Delta \leq k)$-F-*OVERLAY *is* NP*-complete for some k.*

There are several cases in the proof, depending on the structure of graph F. In each case, we give a reduction from (3,4)-SAT problem, which follows the same general idea as the proof of Theorem 41 : we construct variable gadgets H_t containing some negated and non-negated *literal vertices* and identify some of them in such a way that for an assignment ϕ satisfying Φ of (3,4)-SAT problem, $\phi(x_t) = true$ (resp. $false$) if and only if non-negated (resp. negated) literal vertices in the variable gadget are adjacent to w_t in an (F, H, k)-graph.

Lemma 49. *Let F be a graph on p vertices with α_1 vertices of degree λ_1 and $\alpha_2 = p - \alpha_1$ vertices of degree $\lambda_2 > \lambda_1$ such that $F[V_{\lambda_2}]$ is μ-regular but neither complete nor anticomplete. Then there exists k such that $(\Delta \leq k)$-F-*OVERLAY *is* NP*-complete.*

Proof. Set $\gamma = \lambda_2 - \mu$ and $k = \max\{4\gamma(\alpha_2 - 1) + 4\mu + \lambda_1, 3\gamma(\alpha_2 - 1) + 6\mu - 1 + \lambda_1\}$.

We give a reduction from (3,4)-SAT. Given a formula Φ of this problem, we construct a hypergraph H as follows.

1. For each variable x_t, we construct a *variable gadget* H_t in the following way. We first create a *center vertex* w_t, a set of $4(\alpha_2-1)$ vertices $U_t = \{u_t^1, \ldots, u_t^{4(\alpha_2-1)}\}$, and for each $i \in [4(\alpha_2-1)]$, create a set of α_1 new vertices W_t^i, and a hyperedge $S_t^i = W_t^i \cup \{w_t, u_t^i, \ldots, u_t^{i+\alpha_2-2}\}$ (superscripts are modulo $4(\alpha_2-1)$).
   For $r \in [4]$, let $x_t^r = u_t^{r(\alpha_2-1)-\alpha_2+2}$ and $\bar{x}_t^r = u_t^{r(\alpha_2-1)-\alpha_2+3}$. Set $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ and $\overline{X}_t = \{\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \bar{x}_t^4\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated* (resp. *negated) literal vertices* of H_t.
2. For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we identify y_1, y_2, y_3 into a *clause vertex* c_j, where $y_i = x_t^r$ if $\ell_i = x_t$ and ℓ_i is the r-th occurrence of x_t, and $y_i = \bar{x}_t^r$ if $\ell_i = \bar{x}_t$ and is the r-th occurrence of x_t.
3. Finally, we add degree-gadgets on some vertices.
   - We add a $(k - 4\gamma(\alpha_2-1) - 4\mu)$-degree gadget on vertex w_t.
   - We add a $(k-\lambda_1)$-degree gadget at every vertex in W_t^i for all $i \in [4(\alpha_2-1)]$.
   - For $i \in [\mu-1]$, we add a $(k - \gamma(\alpha_2-1) - 2\mu + 1)$-degree-gadget on each $u_t^{(\alpha_2-1)r-j}$ for $r \in [4]$.

Observe that every vertex in W_t^i, for $i \in [4(\alpha_2-1)]$, has degree at least $k-\lambda_1$ in its $(k-\lambda_1)$-degree-gadget, and so degree at most λ_1 in S_t^i. Thus, each of those vertices must have degree λ_1 in S_t^i. It implies that all the other vertices must have degree at least λ_2 in any hyperedge of H_t. In particular, w_t has degree λ_2 in any hyperedge of H_t. Since w_t is in a $(k - 4\gamma(\alpha_2-1) - 4\mu)$-degree-gadget and is adjacent to γ vertices in every W_t^i for $i \in [4(\alpha_2-1)]$, it has degree at most 4μ in $\bigcup_{i=1}^{4(\alpha_2-1)} S_t^i \setminus W_t^i$.

Moreover, each vertex $u_t^i \in U_t$ is a λ_2-vertex in every hyperedge $S_t^{i'}$ of H_t containing it, and so adjacent to γ vertices in $W_t^{i'}$. Since u_t^i belongs to α_2-1 hyperedges of H_t, thus u_t^i is adjacent to $\gamma(\alpha_2-1)$ vertices in $\bigcup_{i'=1}^{4(\alpha_2-1)} W_t^{i'}$. For $i \in [\mu-1]$, $u^{(\alpha_2-1)r-i}$ is in a $(k-\gamma(\alpha_2-1)-2\mu+1)$-degree-gadget, then it has degree at most $\gamma(\alpha_2-1)+2\mu-1$ in H_t, and so at most $2\mu-1$ in $\bigcup_{i'=1}^{4(\alpha_2-1)} S_t^{i'}$. Moreover, $F[V_{\lambda_2}]$ is μ-regular (but not complete or anticomplete). The following is then similar to the proof of Theorem 41 for $F[V_{\lambda_2}]$. So we just sketch it.

Assume that there exists an assignment ϕ satisfying Φ. Let G be the graph obtained as follows.

We let $(v_0, v_1, \ldots, v_{\alpha_2-1})$ be an ordering of V_{λ_2} such that $N_F(v_0) = \{v_{\alpha_2-\mu+1}, \ldots, v_{\alpha_2-1}\} \cup \{v_1\}$ if $\phi(x_t) = true$ and such that $N_F(v_0) = \{v_{\alpha_2-\mu+1}, \ldots, v_{\alpha_2-1}\} \cup \{v_2\}$ if $\phi(x_t) = false$). For every $i \in [4\alpha_2-4]$, we let $G[S_t^i]$ be the copy of F in which every vertex in W_t^i is a λ_1-vertex, w_t corresponds to v_0 and $u_t^{i'}$ for $i' \in \{i, \ldots, i+\alpha_2-1\}$ corresponds to the vertex $v_{i''}$ such that $i' \equiv i'' \mod \alpha_2 - 1$. In addition, for every d-degree-gadget D at some vertex v, we let $G[V(D)]$ be an (F, D, k)-graph in which v has degree d.

The graph G F-overlays H and one can check that $\Delta(F) \leq k$.

Conversely, assume that G is an (F, H, k)-graph. One can prove the following claim.

Claim 49.1. *Every subgraph $G[H_t]$ satisfies the following.*

(a) *Every vertex in W_t^i for $i \in [4\alpha_2 - 4]$ is a λ_1-vertex in S_t^i.*

(b) *w_t is a λ_2-vertex in every hyperedges of H_t. Furthermore, it is adjacent to γ vertices in each W_t^i and the vertices $u_t^{(\alpha_2-1)r-i}$ for $r \in [4]$, $i \in [\mu - 1]$.*

Therefore the truth assignment ϕ defined by $\phi(x_t) = true$ (resp. $false$) if w_t is adjacent to X_t (resp. $\overline{X}_t$), satisfies Φ. □

Proof of Theorem 48. Let V_d be the set of vertices of degree d in F and $\mathbf{d} = (d_1, \dots, d_p)$ be the non-decreasing degree sequence of F. Let N_s be the set of vertices of V_{λ_2} having exactly s neighbors in V_{λ_1}, and let $N_{\geq s} = \bigcup_{s' \geq s} N_{s'}$.

For technical reasons, we distinguish several cases as follows.
If $F[V_{\lambda_1}]$ is not anticomplete, then see **Case A**.
Otherwise, $F[V_{\lambda_1}]$ is anticomplete. Assume first that $F[V_{\lambda_2}]$ is regular. If $F[V_{\lambda_2}]$ is neither complete nor anticomplete, then we have the result by Lemma 49. If $F[V_{\lambda_2}]$ is anticomplete, then F is a disjoint of union of $K_{\lambda_1,\lambda_1+1}$ and we have the result by Theorem 43.
Hence we may assume that $F[V_{\lambda_2}]$ is complete. Observe $\alpha_2 \geq \lambda_1$ because a vertex of V_{λ_1} has all its neighbors in V_{λ_2} and $\alpha_2 \leq \lambda_1 + 1$ because every vertex of V_{λ_2} is adjacent to all other vertices of V_{λ_2} and at least one in V_{λ_1}. If $\alpha_2 = \lambda_1 + 1$, then every vertex of V_{λ_2} has exactly one neighbor in V_{λ_1}, and so $\alpha_2 = \lambda_1 \times \alpha_1$. Hence $\alpha_2 = 2 = \alpha_1$ and $\lambda_1 = 1$. Thus $F = K_3^-$ and we have the result by Theorem 42. If $\alpha_2 = \lambda_1$, then every vertex of V_{λ_1} is adjacent to all vertices of V_{λ_2}. Thus F is $K_{\lambda_1+2}^-$ and we have the result by Theorem 42. Henceforth, we may assume that $F[V_{\lambda_2}]$ is not regular, that is $F[V_{\lambda_2}]$ has at least two degree values. In particular, $\alpha_2 \geq 2$.

If $N_{\geq 2}$ is empty, then $V_{\lambda_2} = N_0 \cup N_1$ and both N_0, N_1 are non-empty. See **Case B**-(i).

If there is a vertex in $N_{\geq 2}$ which is not adjacent to a vertex in V_{λ_1}, see **Case C**-(i).

Otherwise, every vertex in $N_{\geq 2}$ is adjacent to all vertices in V_{λ_1} (so here $N_{\geq 2} = N_{\alpha_1}$ with $\alpha_1 \geq 2$). If $N_1 = \emptyset$, then see **Case C**-(ii). Otherwise, $N_1 \neq \emptyset$ and any vertex in V_{λ_1} is not adjacent to all vertices in N_1, see **Case B**.

<u>Case A:</u> We set k depending on the subgraph $F[V_{\lambda_1}]$ of F.

(1) If $F[V_{\lambda_1}]$ is not complete, then $k = 6\lambda_1 - 1$.

(2) If $F[V_{\lambda_1}]$ is complete, then every vertex of V_{λ_1} is not adjacent to some vertex in V_{λ_2}. We set $k = 6\lambda_1 + 3$.

We give a reduction from (3,4)-SAT .

Given a formula Φ of this problem, we build a hypergraph H as follows.

1. For each variable x_t, we construct a *variable gadget* H_t in the following way. We first create a *center vertex* w_t, a set of $4p - 4$ vertices $U_t = \{u_t^1, \dots, u_t^{4p-4}\}$, and

$4p-4$ hyperedges $S_t^i = \{w_t, u_t^i, \ldots, u_t^{i+p-2}\}$ (superscripts are modulo $4(p-1)$) for $i \in [4p-4]$.

For $r \in [4]$, let $x_t^r = u_t^{r(p-1)-p+2}$ and $\bar{x}_t^r = u_t^{r(p-1)-p+3}$. Set $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ and $\overline{X}_t = \{\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \bar{x}_t^4\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated* (resp. *negated) literal vertices* of H_t.

2. For each variable x_t, we add a set of $p - \lambda_1$ vertices W_t, and a hyperedge $S'_t = W_t \cup \{u_t^{p-1}, \ldots, u_t^{p-\lambda_1+1}\}$ and we add a $(k - 4\lambda_1 - 1)$-degree-gadget at w_t.
3. For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we identify y_1, y_2, y_3 into a *clause vertex* c_j, where for all $i \in [3]$, $y_i = x_t^r$ if $\ell_i = x_t$ and ℓ_i is the r-th occurrence of x_t, and $y_i = \bar{x}_t^r$ if $\ell_i = \bar{x}_t$ and is the r-th occurrence of x_t.

Let z be a vertex in V_{λ_1} which is adjacent to the minimum number $a > 0$ of vertices in this set. Let $(z, z_1, \ldots, z_{p-1})$ be an ordering of F such that :

- z_j has degree λ_1 and is adjacent to z for all $j \in [a]$,
- z_j has degree λ_1 and is not adjacent to z for all $a + 1 \leq j \leq \alpha_1 - 1$,
- z_j has degree λ_2 and is adjacent to z for all $\alpha_1 \leq j \leq \alpha_1 + \lambda_1 - a - 1$.
- z_j has degree λ_2 and is not adjacent to z for all $\alpha_1 + \lambda_1 - a \leq j \leq p - 1$.

We will show that there is an assignment ϕ satisfying Φ if and only there is an (F, H, k)-graph G.

Assume that ϕ satisfies Φ, then we construct G as follows. For all $i \in [4p-4]$, let $G[S_t^i]$ be copies of F such that w_t corresponds to the vertex z and the following hold.

In **Case A**-(1),

- if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$), then each vertex in X_t (resp. $\overline{X}_t$) corresponds to z_1, and each of $\overline{X}_t$ (resp. X_t) corresponds to $z_{\alpha_1 - 1}$.
- for all $r \in [4]$ and $2 \leq i \leq \alpha_1 - 2$, $u^{(p-1)r+1-i}$ corresponds to z_i.
- for all $r \in [4]$ and $\alpha_1 \leq i \leq p - 1$, $u^{(p-1)r+2-i}$ corresponds to z_i.

In **Case A**-(2),

- if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$), then each vertex in X_t (resp. $\overline{X}_t$) corresponds to z_1, and each of $\overline{X}_t$ (resp. X_t) corresponds to z_{p-1}.
- for all $r \in [4]$ and $2 \leq i \leq p - 2]$, $u^{(p-1)r+1-i}$ corresponds to z_i.

For any d-degree-gadget D at a vertex v, let $G[V(D)]$ be an (F, D, k)-graph in which v has degree d.

Let us check that $\Delta(G) \leq k$.

- w_t is adjacent to $4\lambda_1$ vertices in H_t and one more in $W_t \subset V(S'_t)$, and it has degree $k - 4\lambda_1 - 1$ in its degree-gadget. Thus w_t has degree k in total.
- Any literal vertex which is not identified to any clause vertex and is not in S'_t has degree has degree at most $2\lambda_2$ in its variable gadget. So, it has degree less than k.

- Any literal vertex which is in S'_t has degree has degree at most $2\lambda_2$ in its variable gadget and it is adjacent to at most λ_1 vertices in W_t. So, it has degree less than k.
- Each clause vertex c_j is in three literal variable gadgets. In **Case A**-(1) (resp. **Case A**-(2)), c_j has degree at most $2\lambda_1$ (resp. $2\lambda_2$) in each variable gadget. Moreover at least one of the literals is true so its corresponding literal vertex has degree $2\lambda_1 - 1$ (resp. $2\lambda_2 - 1$) Therefore c_j has degree at most $6\lambda_1 - 1$ (resp. $6\lambda_2 - 1$) in its variable gadget. Now it has degree $k - 6\lambda_1 + 1$ (resp. $k - 6\lambda_2 + 1$) in its degree-gadget, and so at most k in total.
- Any vertex which is in a degree-gadget but in no variable gadget has degree at most $2\lambda_t \leq k$ since it belongs to at most two hyperedges.
- Any vertex in W_t has degree at most $\lambda_2 < k$.

Hence, G is an (F, H, k)-graph.

Conversely, let G be an (F, H, k)-graph.

Claim 49.2. *Any $G[H_t]$ satisfies the following.*

(a) *w_t has degree $4\lambda_1$ in H_t and w_t has degree exactly λ_1 in every hyperedge containing it.*

(b) *There is $I \in [p-1]$ of size λ_1 such that for all $i \in I$ and $r \in [4]$, $u_t^{(p-1)r-i+1}$ is adjacent to w_t; and $[\lambda_1 - 1] \subset I$.*

Proof of claim: Observe that w_t is in a $(k - 4\lambda_1 - 1)$-degree-gadget, so it has degree at most $4\lambda_1 + 1$ in $H_t \cup S'_t$. Since w_t is in S'_t which intersects S^1_t in $\lambda_1 - 1$ vertices and it is at least λ_1 in S'_t, then w_t is adjacent to at least one vertex in $W_t \subset V(S'_t)$. Thus, w_t has degree at most $4\lambda_1$ in H_t. Now for every $i \in [p-1]$, w_t belongs to the four hyperedges $S_t^{(p-1)r-i}$, $r \in [4]$, which pairwise intersect in $\{w_t\}$. Hence w_t has degree exactly λ_1 in each $S_t^{(p-1)r-i}$ and then $4\lambda_1$ in H_t. This proves (a).

Now, if a vertex u^i_t is adjacent to w_t, then so is u_t^{i+p-1} because w_t has degree exactly λ_1 in both S^i_t and S^{i+1}_t. Therefore there is $I \in [p-1]$ of size λ_1 such that w_t is adjacent to $u_t^{(p-1)r-i+1}$ for all $i \in I$ and $r \in [4]$.
Since w_t has degree $4\lambda_1$ in H_t, then w_t is adjacent to exactly one vertex in W_t and so must be adjacent to $\lambda_1 - 1$ vertices in $S'_t \setminus W_t$ which are $u_t^{p-1}, \ldots, u^{p-\lambda_1+1}$. It implies that $[\lambda_1 - 1] \subset I$. This proves (b). ◁

Claim 49.2 implies that the vertices of X_t (resp. $\overline{X}_t$) are either all adjacent to w_t or all non-adjacent to w_t. Moreover, w_t is adjacent to $\lambda_1 - 1$ vertices not in $X_t \cup \overline{X}_t$. Hence if the vertices of X_t are adjacent to w_t, the vertices of $\overline{X}_t$ are not (and vice-versa).

Let ϕ be the truth assignment defined by $\phi(x_t) = true$ if w_t is adjacent to H_t, and $\phi(x_t) = false$ otherwise. In any clause vertex c_j, we identified three literal vertices corresponding to the three literals.

- In **Case A**-(1), c_j has degree at most $k = 6\lambda_1 - 1$, so it has degree less than $2\lambda_1$ in one of its three variable gadgets H_t. Since any vertex u^i_t for $i \in [4p-4]$ belongs to two hyperedges S_t^i and S_t^{i-p+2} which intersect in $\{u_t^i, w_t\}$ and has degree at least λ_1 in each, then it has degree $2\lambda_1 - 1$ in H_t only if it is adjacent to w_t. Hence, c_j is adjacent to w_t.

- In **Case A**-(2), c_j has degree at most $k = 6\lambda_1 + 3 < 6\lambda_2$, so it has degree less than $2\lambda_2$ in one of its three variable gadgets H_t.
  Moreover, $F[V_{\lambda_1}]$ is complete, then w_t is adjacent to all λ_1-vertices in every hyperedges of H_t (because it is a λ_1-vertex in every hyperedge of H_t). If c_j is not adjacent to no center vertex of the three variable gadgets it belongs to, then it must be a λ_2-vertex in each hyperedge of those gadgets. Thus it has degree at least $2\lambda_2$ in each variable gadget and so at least $6\lambda_2$ in total, a contradiction. Thus c_j is adjacent to the center of at least one variable gadget w_t.

Hence, the corresponding literal to the literal vertex adjacent to w_t for variable x_t is true and clause C_j is satisfied.

Consequently, ϕ satisfies Φ.

Case B: Recall that in that case $N_1 \neq \emptyset$. Let $\gamma = \max\{|N(v) \cap V_{\lambda_1}| \mid v \in V_{\lambda_2}\}$. We have $V_{\lambda_2} = \bigcup_{s=0}^{\gamma} N_s$. Let k as follows.

(i) If $N_0 \neq \emptyset$, then set $k = \max\{6\lambda_2 - 1 + \lambda_1, \gamma\alpha_2 + 2(\lambda_2 - \gamma) + \lambda_1\}$.

(ii) If $N_0 = \emptyset$, $N_{\geq 2} \neq \emptyset$ and every vertex of $N_{\geq 2}$ is adjacent to all vertices of V_{λ_1}, then set $k = \max\{6\lambda_1 + 3\alpha_2 - 1 + \lambda_1, \gamma\alpha_2 + 2(\lambda_2 - \gamma) + \lambda_1\}$. Note that in that case every vertex in V_{λ_1} is adjacent to a vertex in N_1 but not all.

We give a reduction from (3,4)-SAT.

Given a formula Φ of this problem, we build a hypergraph as follows.

1. For each variable x_t, we construct a *variable gadget* H_t in the following way.
   We first create a *center vertex* w_t, $4\alpha_2$ sets of $\alpha_1 - 1$ vertices A_t^i for $i \in [4\alpha_2]$, a set of $4\alpha_2$ vertices $U_i = \{u_t^1, \ldots, u_t^{4\alpha_2}\}$, and $4\alpha_2$ hyperedges $S_t^i = A_t^i \cup \{w_t, u_t^i, \ldots, u_t^{i+\alpha_2-1}\}$ (superscripts are modulo $4\alpha_2$) for $i \in [4\alpha_2]$.
   For $r \in [4]$, let $x_t^r = u_t^{\alpha_2(r-1)+1}$ and $\bar{x}_t^r = u_t^{\alpha_2(r-1)+2}$. Set $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ and $\overline{X}_t = \{\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \bar{x}_t^4\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated* (resp. *negated) literal vertices* of H_t.
2. For each variable x_t,
   - we create a set of $p - \lambda_1$ vertices B_t and a hyperedge $S'_t = B_t \cup \{w_t, u_t^{\alpha_2}, \ldots, u_t^{\alpha_2-\lambda_1+2}\}$.
   - add a $(k - 4\lambda_1 - 1)$-degree-gadget on w_t.
   - add a $(k - \lambda_1)$-degree-gadget on every vertex in A_t^i for $i \in [4\alpha_2]$.
3. For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we identify y_1, y_2, y_3 into a *clause vertex* c_j, where $y_i = x_t^r$ if $\ell_i = x_t$ and ℓ_i is the r-th occurrence of x_t, and $y_i = \bar{x}_t^r$ if $\ell_i = \bar{x}_t$ and is the r-th occurrence of x_t. We also add a $(k - 6\lambda_2 - 1)$-degree vertex at c_j in **Case B**-(i), and a $(k - 6\lambda_1 - 3\alpha_2 + 1)$-degree vertex at c_j in **Case B**-(ii).

We will show that there is an assignment ϕ satisfying Φ if and only there is an (F, H, k)-graph G.

Let z be a vertex in V_{λ_1} adjacent to a vertex y in N_1 and let $\bar{y}$ be a vertex in N_0 in **Case B**-(i) or a vertex in N_1 not adjacent to z in **Case B**-(ii). Note that $\bar{y}$ and z are not adjacent. Let $(y_1, \ldots, y_{\alpha_2-2})$ be an ordering of $V_{\lambda_2} \setminus \{y, \bar{y}\}$ such that $y_1, \ldots, y_{\lambda_1-1}$ are adjacent to z and $y_{\lambda_1}, \ldots, y_{\alpha_2-2}$ are not adjacent to z.

Assume that there is ϕ satisfying Φ, we construct a graph G as follows. Let $G[S'_t]$ be a copy of F such that w_t has degree λ_1 and is adjacent to the $\lambda_1 - 1$ vertices $u_t^{\alpha_2}, \ldots, u_t^{\alpha_2-\lambda_1+2}$.
In a variable gadget H_t, for every $i \in [4\alpha_2]$, we let $G[S_t^i]$ be a copy of F such that w_t corresponds to the vertex z, and

- A_t^i corresponds to $V_{\lambda_1} \setminus \{z\}$.
- if $\phi(x_t) = true$ (resp. $\phi(x_t) = false$), then each vertex in X_t (resp. $\overline{X}_t$) corresponds to y, and each vertex in $\overline{X}_t$ (resp. X_t) to $\bar{y}$.
- for $i \in [\alpha_2 - 2]$, $u_t^{(p-1)r-i+1}$ corresponds to y_i.

For any d-degree-gadget D at a vertex v, let $G[V(D)]$ be an (F, D, k)-graph in which v has degree d.

Let us check that $\Delta(G) \leq k$.

- w_t is adjacent to $4\lambda_1$ vertices in H_t and one more in $W_t \subset V(S'_t)$, and it has degree $k - 4\lambda_1 - 1$ in its degree-gadget, then w_t has degree k in total.
- Any literal vertex which is not identified to any clause vertex and not in S'_t has degree at most $\gamma\alpha_2 + 2(\lambda_2 - \gamma)$ in its variable gadget (it is adjacent to at most γ vertices in each A_t^i in a hyperedge to which it belongs and there are α_2 such hyperedges; and $f(x) = x\alpha_2 + 2(\lambda_2 - x)$ is increasing). So, it has degree less than k.
- Any literal vertex which is not identified to any clause vertex and in S'_t has degree at most $\gamma\alpha_2 + 2(\lambda_2 - \gamma)$ in its variable gadget and is adjacent to at most λ_1 vertices in B_t. So it has degree at most k.
- Each clause vertex c_j is in three variable gadget. In **Case B**-(i), c_j (resp. **Case B**-(ii)), in each of these gadgets, c_j has degree either $2\lambda_2 - 1$ if it is adjacent to w_t or $2\lambda_2$ (resp. $\alpha_2 + 2(\lambda_2 - 1)$ otherwise. Moreover at least one of the literals is true, its corresponding literal vertex has degree $2\lambda_2 - 1$ in its variable gadget. Therefore c_j at most $6\lambda_2 - 1$ neighbors (resp. $2\alpha_2 + 6(\lambda_2 - 1) + 1$) in variable gadgets. It also has $k - 6\lambda_2 + 1$ (resp. $k - 6\lambda_1 - 3\alpha_2 + 1$) neighbors in its degree-gadget. Hence, in G, it has degree at most k.
- Any vertex which is in a degree-gadget but in no variable gadget has degree at most $2\lambda_t \leq k$ since it belongs to at most two hyperedges.
- Any vertex in B_t has degree at most $\lambda_2 < k$.

Hence, G is an (F, H, k)-graph.

Conversely, let G be an (F, H, k)-graph.
Observe that any vertex in A_t^i for $i \in [4\alpha_2]$ is in a $(k - \lambda_1)$-degree-gadget, then it has

degree at most λ_1 in S_t^i. Since any vertex has degree at least λ_1 in a hyperedge, then every vertex in $\bigcup_{i=1}^{4\alpha_2} A_t^i$ is a λ_1-vertex in any hyperedge to which it belongs.

Claim 49.3. *Any $G[H_t]$ satisfies the following.*

(a) *w_t has degree $4\lambda_1$ in H_t and w_t is a λ_1-vertex in every hyperedge of H_t.*

(b) *There is $I \in [\alpha_2]$ of size λ_1 such that for all $i \in I$ and $r \in [4]$, $u_t^{\alpha_2 r - i+1}$ is adjacent to w_t; and $[\lambda_1 - 1] \subset I$.*

This claim can be proved in exactly the same way as Claim 49.2.

We have that every vertex in U_t is a λ_2-vertex in any hyperedge to which it belongs (since w_t and $\alpha_1 - 1$ vertices of A_t^i for $i \in [4\alpha_2]$ are λ_1-vertices).
Claim 49.3 implies that the vertices of X_t (resp. $\overline{X}_t$) are either all adjacent to w_t or all non-adjacent to w_t. Moreover, w_t is adjacent to $\lambda_1 - 1$ vertices in U_t but not in $X_t \cup \overline{X}_t$. Hence if the vertices of X_t are adjacent to w_t, the vertices of $\overline{X}_t$ are not (and vice-versa).

Let ϕ be the truth assignment defined by $\phi(x_t) = true$ if w_t is adjacent to all vertices of X_t in H_t, and $\phi(x_t) = false$ otherwise.
A clause vertex c_j has degree at most k. Because of its degree-gadget, in **Case B**-(i) (resp. **Case B**-(ii)), it has degree at most $6\lambda_2 - 1$ (resp. $6\lambda_1 + 3\alpha_2 - 1$) in H_t. Now, since it is the identification of three literal vertices, c_j has degree less than $2\lambda_2$ (resp. $2\lambda_1 + \alpha_2$) in one variable gadget H_t.

Claim 49.4. *Let $i \in [4\alpha_2]$. If u_t^i is not adjacent to w_t, then the following holds.*

(i) *u_t^i has degree at least $2\lambda_2$ in $G[H_t]$;*

(ii) *If $N_0 = \emptyset$, then u_t^i has degree at least $2\lambda_1 + \alpha_2$ in $G[H_t]$;*

Proof of claim: u_t^i has at least λ_2 neighbors in each of S_t^i and $S_t^{i-\alpha_1+1}$. But the intersection of those hyperedges is $\{w_t, u_t^i\}$. As it is not adjacent to w_t, u_t^i has at least $2\lambda_2$ neighbors in $S_t^i \cup S_t^{i-\alpha_1+1}$. This proves (i).

If $N_0 = \emptyset$, then for all $i - \alpha_2 + 1 < i' < i$. u_t^i must be adjacent to at least one λ_1-vertex of $S_t^{i'}$ which is in $A_t^{i'}$. Hence u_i^t has at least $\alpha_2 - 2$ in $\bigcup_{i-\alpha_2+1<i'<i} A_t^{i'}$ which is disjoint from $S_t^i \cup S_t^{i-\alpha_1+1}$. Hence u_t^i has degree at least $2\lambda_2 + \alpha_2 - 2 = 2\lambda_1 + \alpha_2$ in $G[H_t]$. This proves (ii). $\triangleleft$

This claim implies that there is at least one variable gadget H_t in which c_j is adjacent to w_t. It implies that the corresponding literal of this vertex in C_j is true, and so C_j is satisfied.
Consequently, ϕ satisfies Φ.

Case C: In this case, $F[V_{\lambda_1}]$ is anticomplete, $F[V_{\lambda_2}]$ is not regular, and V_{λ_2} satisfies one of the following.

(i) there is a vertex of $N_{\geq 2}$ that is not adjacent to all vertices in V_{λ_1} in F.

(ii) $V_{\lambda_2} = N_{\geq 2} \cup N_0$ and every vertex of $N_{\geq 2}$ is adjacent to all vertices of V_{λ_1}. Since $F[V_{\lambda_2}]$ is not complete, then there is a vertex in $N_{\geq 2}$ which is not adjacent to a vertex in either N_0 **Case C**-(ii)-a or $N_{\geq 2}$ **Case C**-(ii)-b.

We set $a = \max_{\substack{u \in V_{\lambda_1} \\ v \in N(u)}} \left|N(v) \cap N(u)\right|$, and let $k = 4(p-2)(2\lambda_1 - a) + 4\lambda_1 + 1$.

For conveniences, we denote some vertices of graph F as follows. Let z_0 be a vertex in $N_{\geq 2}$ such that there is $z_1 \in V_{\lambda_1}$ adjacent to z_0 with $a = |N(z_0) \cap N(z_1)|$. Let $z \in N_{\geq 2}$ which is adjacent to the minimum number of vertices in N_0, and $y, y' \in V_{\lambda_1}$ be vertices adjacent to z and

- in **Case C**-(i), let $\bar{y} \in V_{\lambda_1}$ be a vertex not adjacent to z.
- in **Case C**-(ii), let $\bar{y}$ be a vertex not adjacent to z such that $\bar{y} \in N_0$ if z is not adjacent to all vertices in N_0 and $\bar{y} \in N_{\geq 2}$ otherwise.

We give a reduction from (3,4)-SAT.

Given a formula Φ of this problem, we build a hypergraph as follows.

1. For each variable x_t, we construct a *variable gadget* H_t in the following way. We first create a *center vertex* w_t, a set of $4(p-2)$ vertices $D_t = \{d_t^1, \ldots, d_t^{4(p-2)}\}$, a set of $4(p-2)$ vertices $U_t = \{u_t^1, \ldots, u_t^{4(p-2)}\}$, and $4(p-2)$ hyperedges $S_t^i = \{w_t, d_t^i, u_t^i, \ldots, u_t^{i+p-3}\}$ (superscripts are modulo $4(p-2)$) for $i \in [4(p-2)]$. For $r \in [4]$, let $x_t^r = u_t^{r(p-2)-p+3}$ and $\bar{x}_t^r = u_t^{r(p-2)-p+4}$. Set $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ and $\overline{X}_t = \{\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \bar{x}_t^4\}$. The vertices of X_t (resp. $\overline{X}_t$) are called the *non-negated* (resp. *negated) literal vertices* of H_t.
2. For each variable x_t,
   - We create a set Y_t of $p - \lambda_1$ vertices and a hyperedge $S'_t = Y_t \cup \{w_t\} \cup \{u_t^{p-2}, \ldots, u_t^{p-\lambda_1}\}$.
   - For any $i \in [4(p-2)]$, we add two sets of $p - \lambda_1 - 1$ vertices A_t^i, B_t^i and a set of $\lambda_1 - 1$ vertices C_t^i, and two hyperedges $A_t^i \cup C_t^i \cup \{d_t^i, w_t\}$ and $B_t^i \cup C_t^i \cup \{d_t^i, w_t\}$. We call this a *fickle-gadget* F_t^i.
   - We add a $(k - 2\lambda_1 + 1)$-degree-gadget on every vertex in D_t.
3. For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, we identify y_1, y_2, y_3 into a *clause vertex* c_j, where $y_i = x_t^r$ if $\ell_i = x_t$ and ℓ_i is the r-th occurrence of x_t, and $y_i = \bar{x}_t^r$ if $\ell_i = \bar{x}_t$ and is the r-th occurrence of x_t. We also add at c_j a $(k - 6\lambda_1 + 1)$-degree-gadget in **Case C**-(i), a $(k - 6\lambda_1 - 3)$-degree-gadget in **Case C**-(ii)-a, and a $(k - 6\lambda_1 - 3p + 6)$-degree-gadget in **Case C**-(ii)-b.

We will show that there is an assignment ϕ satisfying Φ if and only if there is an (F, H, k)-graph G.

Assume that there is ϕ satisfying Φ, we construct a graph G as follows.

Let $(y_1 \ldots, y_{p-4})$ be an ordering of $V(F) \setminus \{z, y, y', \bar{y}\}$ such that $y_1, \ldots, y_{\lambda_1 - 2}$ are adjacent to z and $y_{\lambda_1 - 1}, \ldots, y_{p-4}$ are not adjacent to z.

In both hyperedges of any fickle-gadget F_t^i, w_t corresponds to z_0 and d_t^i corresponds z_1. d_t^i is adjacent to w_t and all vertices in C_t^i, while w_t is adjacent to a vertices in C_t^i and $\lambda_2 - a$ other ones in each of A_t^i, B_t^i.
Let $G[S_t']$ be a copy of F such that w_t has degree λ_1 and is adjacent to the $\lambda_1 - 1$ vertices $u_t^{p-2}, \ldots, u_t^{p-\lambda_1}$.

For each variable gadget H_t, for every $i \in 4[\alpha_2]$, let $G[S_t^i]$ be a copy of F such that w_t corresponds to z, and

- if $\phi(x_t) = true$ (resp. $false$), then each vertex of X_t (resp. $\overline{X}_t$) corresponds to y and each vertex of $\overline{X}_t$ (resp. X_t) corresponds to $\bar{y}$.
- in any S_t^i, d_t^i corresponds to y'.
- for $i \in [p-4]$ and $r \in [4]$, $u_t^{(p-2)r+1-i}$ corresponds to y_i.

For any d-degree-gadget D at a vertex v, let $G[V(D)]$ be an (F, D, k)-graph in which v has degree d.

Let us check that G has degree at most k.

- Any vertex $d_t^i \in D_t$ has degree $(k - 2\lambda_1 + 1)$ in its degree-gadget and λ_1 in the fickle-gadget F_t^i and $\lambda_1 - 1$ other vertices in $V(S_t^i \setminus \{w_t\})$, thus it has degree k.
- Any vertex in $Y_t \subset V(S_t')$ has degree at most λ_2.
- w_t has $(2\lambda_1 - a)$ neighbor in each of the $4(p-2)$ fickle-gadgets and is adjacent to $4\lambda_1$ vertices in U_t and one more in Y_t. Thus it has degree k in G.
- Any vertex in a degree-gadget which is not in H_t has degree at most $2\lambda_2$.
- Any vertex in U_t but not in $X_t \cup \overline{X}_t \cup S_t'$ has degree at most $2\lambda_2$ if it is not adjacent to any vertex in D_t or at most $2\lambda_2 + p - 2$ if adjacent to a vertex in D_t for each hyperedge to which it belongs.
- Any vertex in $U_t \cap S_t'$ has degree at most $2\lambda_1 + p - 2$ in H_t and it is adjacent to at most λ_1 vertices in Y_t, so it has degree less than k.
- Any clause vertex c_j has degree d in its d-degree-gadget. Moreover, in each of its variable gadget, c_j has degree either $2\lambda_1 - 1$ if it adajcent to the center vertex or $2\lambda_1$ in **Case C**-(i), $2\lambda_2$ in **Case C**-(ii)-a, and $2\lambda_1 + p - 2$ in **Case C**-(ii)-b otherwise. Since there at least one of three literals of the clause C_j is true, c_j has $2\lambda_1 - 1$ in one of its variable gadget, and thus degree at most k in total.

Hence, G is an (F, H, k)-graph.

Conversely, let G be an (F, H, k)-graph.

Claim 49.5. *For any variable x_t, we have the following.*

(a) *For all $i \in [4(p-2)]$, d_t^i is adjacent to w_t and has degree λ_1 in any hyperedge of $F_t^i \cup S_t^i$.*
(b) *w_t is a λ_2-vertex in every hyperedge of H_t. Furthermore, there is $I \in [p-2]$ of size λ_1 such that for all $i \in I$ and $r \in [4]$, $u^{(p-2)r-i+1}$ is adjacent to w_t and $[\lambda_1 - 1] \subset I$.*
(c) *w_t has degree k in G.*

Proof of claim: Observe that any vertex $d_t^i \in X_t$ is in a $(k-2\lambda_1+1)$-degree-gadget, then it has degree at most $2\lambda_1 - 1$ in $G[F_t^i \cup S_t^i]$. Since S_t^i and F_t^i intersect only in $\{d_t^i, w_t\}$ and d_t^i has degree at least λ_1 in each, then d_t^i has degree at least $2\lambda_1 - 1$ in $G[F_t^i \cup S_t^i]$. The equality holds when d_t^i is adjacent to w_t and $\lambda_1 - 1$ vertices in C_t^i. Thus, d_t^i has degree λ_1 in all hyperedges of $F_t^i \cup S_t^i$ and is adjacent to w_t. This proves (a).

In any fickle-gadget F_t^i, from (a), every vertex in $\{w_t\} \cup C_t^i$ is adjacent to d_t^i and must be a λ_2-vertex in the two hyperedges of F_t^i. Thus, w_t is adjacent to at most a vertices in C_t^i, and so has degree at least $2\lambda_2 - a$ in $G[F_t^i]$. Since w_t is in $4(p-2)$ fickle-gadgets, then it has degree at least $4(p-2)(2\lambda_2 - a)$ in $G[\bigcup_{i=1}^{4(p-2)} F_t^i]$.
Moreover, from (a), for $i \in [4(p-2)]$, w_t is adjacent to d_t^i which has degree λ_1 in S_t^i. Thus w_t must be a λ_2-vertex in S_t^i because F is anticomplete.
Since w_t is in S_t' which intersects H_t in $\lambda_1 - 1$ vertices and w_t must have degree at least λ_1 in $G[S_t']$, then it is adjacent to at least one vertex in $Y_t \subset V(S_t')$. Therefore, w_t is adjacent to at most $k - 4(p-2)(2\lambda_2 - a) - 1 = 4\lambda_1$ vertices in H_t.
Now for every $i \in [p-2]$, w_t belongs to four hyperedges $S_t^{(p-2)r-i}$, $r \in [4]$, which pairwise intersect in $\{w_t\}$. Hence w_t has degree exactly λ_1 in each $S^{(p-2)r-i} \setminus \{d_t^{(p-2)r-i}\}$ and then is adjacent to $4\lambda_1$ vertices in U_t.

If a vertex u_t^i is adjacent to w_t, then so is u_t^{i+p-2} because w_t has degree exactly λ_1 in both $S_t^i \setminus \{d_t^i\}$ and $S_t^{i+1} \setminus \{d_t^{i+1}\}$. Therefore there is $I \in [p-2]$ of size λ_1 such that w_t is adjacent to $u_t^{(p-2)r-i+1}$ for all $i \in I$ and $r \in [4]$.
Since w_t has degree $4\lambda_1$ in H_t, then w_t is adjacent to exactly one vertex in Y_t and so must be adjacent to $\lambda_1 - 1$ vertices in $S_t' \setminus Y_t$ which are $u_t^{p-2}, \ldots, u_t^{p-\lambda_1}$. It implies that $[\lambda_1 - 1] \subset I$. This completes the proof of (b).

From (a), (b) we have that w_t has degree $4(p-2)(2\lambda_2 - a)$ in $G[\bigcup_{i=1}^{4(p-2)} F_t^i]$, it is adjacent to $4\lambda_1$ vertices in U_t and one in Y_t. Thus, w_t has degree k in total. This proves (c). $\triangleleft$

Claim 49.5(b) implies that the vertices of X_t (resp. $\overline{X}_t$) are either all adjacent to w_t or all non-adjacent to w_t. Moreover, w_t is adjacent to $4(\lambda_1 - 1)$ vertices in $U_t \setminus (X_t \cup \overline{X}_t)$. Hence if the vertices of X_t are adjacent to w_t, the vertices of $\overline{X}_t$ are not (and vice-versa).

Let ϕ be the truth assignment defined by $\phi(x_t) = true$ if w_t is adjacent to all vertices of X_t in H_t, and $\phi(x_t) = false$ otherwise. Observe the following.

- In **Case C**-(i), any clause vertex c_j is in a $(k - 6\lambda_1 + 1)$-degree-gadget, so it has degree at most $6\lambda_1 - 1$ in the union of its three variable gadgets. Thus it has degree less than $2\lambda_1$ in one of its variable gadgets H_t. Since any vertex in U_t has degree at least $2\lambda_1 - 1$ in $G[H_t]$, with equality only if it is adjacent to w_t, the vertex c_j is adjacent to w_t. Hence, the corresponding literal to this literal vertex is true and so C_j is satisfied.

- In **Case C**-(ii), any clause vertex c_j is in a $(k-d)$-degree-gadget, then has degree at most d in the union of its three variable gadgets. Hence c_j has degree at most $\lfloor d/3 \rfloor$ neighbors in one of those variable gadget, say H_t. Let i be the index such that $c_j = u_t^i$.

  Suppose for a contradiction that c_j is not adjacent to w_t. Then it is a λ_2-vertex in every hyperedge of H_t.
  Vertex c_j has at least λ_2 neighbors in each of S_t^i and S_t^{i-p-3} which intersect in $\{c_j, w_t\}$. Hence c_j has at least $2\lambda_2$ neighbors in $S_t^i \cup S^{i-p-3}$. In **Case C**-(ii)-a, $\lfloor d/3 \rfloor = 2\lambda_1 + 1 < 2\lambda_2$, so we get a contradiction.
  In **Case C**-(ii)-b, since w_t is adjacent to all d_t^i by Claim 49.5 (b), c_j corresponds to a vertex in $N_{\geq 2}$ in every hyperedge of H_t to which it belongs. Therefore it is adjacent to all λ_1-vertices in these hyperedges and thus in particular to all $d_t^{i'}$ for all $i - p + 3 < i' < i$. Hence c_j has degree at least $2\lambda_2 + p - 4$ in $V(H_t)$. But $\lfloor d/3 \rfloor = 2\lambda_1 + p - 3 = 2\lambda_2 + p - 5$, a contradiction.

  In both subcases, the vertex c_j is adjacent to w_t. Hence, the corresponding literal to this literal vertex is true and so C_j is satisfied.

Consequently, ϕ satisfies Φ. □

4.4 On neat hypergraphs

In this section, we consider the problem on a *neat hypergraph* which is a hypergraph in which any two hyperedges intersect in at most one vertex. We reuse the notions of degree sequence $\mathbf{d}$ of a graph F and the multiplicity numbers of degrees in $\mathbf{d}$. The restriction of $(\Delta \leq k)$-F-OVERLAY to neat hypergraphs is called NEAT $(\Delta \leq k)$-F-OVERLAY.

4.4.1 Equivalence with NEAT $(\mathbf{d}, k)$-SPLITTING

Let $\mathbf{d} = (d_1, \dots, d_p)$ be p-uple of positive non-decreasing integers and k be a positive integer. Given a p-uniform hypergraph H, a $(\mathbf{d}, k)$*-splitting* is a set $\boldsymbol{\partial} = \{\partial_S \mid S \in E(H)\}$ of bijections $\partial_S : S \to \mathbf{d}$, such that for every $u \in V(H)$

$$\Sigma_{\boldsymbol{\partial}}(u) = \sum_{S \ni u} \partial_S(u) \leq k.$$

We thus have the following problem :

$(\mathbf{d}, k)$-SPLITTING
Input: A hypergraph H.
Question: Does there exists a $(\mathbf{d}, k)$-splitting of H ?

We call the restriction of $(\mathbf{d}, k)$-SPLITTING to neat hypergraphs NEAT $(\mathbf{d}, k)$-SPLITTING. If $\mathbf{d}$ is a sequence of non-negative integers, we define $\mathrm{np}(\mathbf{d})$ (resp. $\mathrm{np}'(\mathbf{d})$) as the minimum integer k_0 such that $(\mathbf{d}, k)$-SPLITTING (resp. NEAT $(\mathbf{d}, k)$-SPLITTING) is NP-complete for all $k \geq k_0$ or $+\infty$ if no such k_0 exists.

Lemma 50. *Let F be a graph with degree sequence $\mathbf{d}$, let H be a neat $|F|$-uniform hypergraph, and let k be a positive integer. There is an (F, H, k)-graph if and only if there is a $(\mathbf{d}, k)$-splitting of H.*

Proof. Assume first that there is an (F, H, k)-graph G. Since H is neat, free to remove some edges, we may assume that $G[S]$ is a copy of F for all hyperedge S. Hence the mapping d_S defined by $d_S(v) = d_{G[S]}(v)$ is a bijection $S \to \mathbf{d}$. Now, for each vertex v,

$$\sum_{S \ni v} d_S(v) = \sum_{S \ni v} d_{G[S]}(v) = d_G(v) \leq k$$

because G is an (F, H, k)-graph. Thus $\{d_S, \mid S \in E(H)\}$ is a $(\mathbf{d}, k)$-splitting of H.

Reciprocally, assume that there is a $(\mathbf{d}, k)$-splitting $\mathcal{D} = \{d_S, \mid S \in E(H)\}$ of H. For every hyperedge S, take a copy of F_S of F such that $d_{F_S}(v) = d_S(v)$. Let G be the union of the F_S. By construction, G F-overlays H. Moreover, since H is neat, for each vertex v,

$$d_G(v) = \sum_{S \ni v} d_{G[S]}(v) = \sum_{S \ni v} d_{F_S}(v) = \sum_{S \ni v} d_S(v) \leq k$$

because $\mathcal{D}$ is a $(\mathbf{d}, k)$-splitting. Hence G is an (F, H, k)-graph. □

Corollary 51. *If two graphs F_1 and F_2 have the same degree sequence $\mathbf{d}$, then* NEAT $(\Delta \leq k)$-F_1-OVERLAY *and* NEAT $(\Delta \leq k)$-F_2-OVERLAY *have the same complexity, the one of* NEAT $(\mathbf{d}, k)$-SPLITTING.

Remark 52. Note that (NEAT) $(\mathbf{d}, k)$-SPLITTING is also defined for sequences $\mathbf{d}$ which are not degree sequences. In particular, we might allow the elements of $\mathbf{d}$ to be negative. However, we shall restrict ourselves to *non-negative* sequences, which are sequences all elements of which are non-negative..

Let $\mathbf{d} = (d_1, \ldots, d_p)$ and $\mathbf{d}' = (d'_1, \ldots, d'_{p'})$ be two sequences. The *concatenation* of $\mathbf{d}$ and $\mathbf{d}'$, denoted by $\mathbf{d} \cdot \mathbf{d}'$, is the sequence of $p + p'$ elements of d and d' in non-decreasing order. We denote by $\mathbf{0}^p$ the sequence of p elements equal to 0. A degree sequence is *potentially connected* if there is a connected graph G with degree sequence $\mathbf{d}$.

Theorem 53 (Folklore). *Let degree sequence $(d_1, \ldots, d_n)$ is potentially connected if and only if $d_1 \geq 1$ and $\sum_{i=1}^{n} d_i \geq 2(n-1)$.*

Many results obtained from the previous sections for $(\Delta \leq k)$-F-OVERLAY can be adapted to NEAT $(\Delta \leq k)$-F-OVERLAY and $(\mathbf{d}, k)$-SPLITTING. For F with isolated vertices and F' is the induced subgraph of F by non-isolated vertices, Corollary 36 gives a sufficient condition in which the NP-completeness of $(\Delta \leq k)$-F'-OVERLAY implies that of $(\Delta \leq k)$-F-OVERLAY. It is also true for $(\mathbf{d}, k)$-SPLITTING and even stronger on neat hypergraphs.

Corollary 54. *Let F be a graph, $\mathbf{d}$ a non-negative sequence, and q a positive integer.*

(i) *If $(\mathbf{d}, k)$-SPLITTING is NP-complete, then $(\mathbf{0}^q \cdot \mathbf{d}, k)$-SPLITTING is also NP-complete.*

(ii) *If NEAT $(\Delta \leq k)$-F-OVERLAY is NP-complete, then $(\Delta \leq k)$-$(F + E_q)$-OVERLAY is also NP-complete.*

4.4.2 Polynomial results

Observe that for F an r-regular graph, NEAT $(\Delta \leq k)$-F-OVERLAY is simple. Since a hyperedge intersects another in at most one vertex in a neat hypergraph, then the total degree of a vertex v in a graph G F-overlays H must be $rd_H(v)$. Therefore, if the maximum degree of a vertex in H is at most $\lfloor \frac{k}{r} \rfloor$, then there exists an (F, H, k)-graph for the problem. We then provide a proof when F has two degree values.

Theorem 55. *Let F be a graph with degree values $\lambda_1 < \lambda_2$ and k be a non-negative integer.* NEAT $(\Delta \leq k)$-F-OVERLAY *is in* P.

Proof. Let H be a neat $|F|$-uniform hypergraph.

If there is a vertex v such that $\lambda_1 \cdot d_H(v) > k$, then we answer 'No'. Indeed in an (F, H, k)-graph G, the vertex v has degree at least λ_1 in each subgraph of G induced by a hyperedge to which it belongs, and so degree at least $\lambda_1 \cdot d_H(v)$ in G because H is neat. Hence, we assume that $\lambda_1 \cdot d_H(v) \leq k$ for every vertex.

Let $B(H)$ be the incidence graph of H: its vertex set is $E(H) \cup V(H)$ and there is an edge between a hyperedge $S \in E(H)$ and a vertex $v \in V(H)$ if and only if $v \in S$.

Claim 55.1. *There is an (H, F, k)-graph if and only if $B(H)$ has a subgraph in which every hyperedge vertex has degree α_2 and every vertex has degree at most $\lfloor \frac{k - \lambda_1 \cdot d_H(v)}{\lambda_2 - \lambda_1} \rfloor$.*

Proof of claim: Assume that there is an (H, F, k)-graph G. Observe that since H is a neat hypergraph, free to remove some edges, we may assume that $G[S]$ is isomorphic to F for every hyperedge S of H. Let B' be the subgraph of $B(H)$ in which a vertex v is adjacent to a hyperedge S if and only if v has degree λ_2 in S. By definition every hyperedge has degree α_2 in B'. Consider now a vertex v. Its degree in G is the sum of its degree in the subgraphs induced by the hyperedges S to which it belongs because H is neat. Hence $d_G(v) = \lambda_2 \cdot d_{B'}(v) + \lambda_1(d_H(v) - d_{B'}(v)) = \lambda_1 \cdot d_H(v) + (\lambda_2 - \lambda_1) d_{B'}(v)$. But G is an (F, H, k)-graph, so $d_G(v) \leq k$. With the previous equality, this yields $d_{B'}(v) \leq \frac{k - \lambda_1 \cdot d_H(v)}{\lambda_2 - i_1}$, and so $d_{B'}(v) \leq \lfloor \frac{k - \lambda_1 \cdot d_H(v)}{\lambda_2 - \lambda_1} \rfloor$.

Conversely, assume $B(H)$ has a subgraph B' in which every $S \in E(H)$ has degree α_2 and any $v \in V(H)$ has degree at most $\lfloor \frac{k-\lambda_1 \cdot d_H(v)}{\lambda_2-\lambda_1} \rfloor$. We construct G as follows: for each hyperedge S, the induced subgraph $G[S]$ is a copy of F such that any vertex v where $(S,v) \in E(B')$ is of degree λ_2 (there are α_2 such vertices) and the remaining vertices have degree λ_1. Clearly, G F-overlays H. Moreover every vertex $v \in V(H)$ has degree λ_2 in at most $\lfloor \frac{k-\lambda_1 \cdot d_H(v)}{\lambda_2-\lambda_1} \rfloor$ subgraphs and degree λ_1 in the remaining subgraphs induced by hyperedges to which it belongs. Hence v has degree at most k in G. Therefore G is an (F,H,k)-graph. $\lhd$

Now the existence of a subgraph of $B(H)$ as shown in Claim 55.1 can be done in polynomial time (see [HHKL90] Chap. 10). $\square$

By Corollary 51, we obtain the following.

Corollary 56. *Let $\mathcal{F}$ be a family of graphs with the same degree sequence such that it has only one value λ or exactly two degree values $\lambda_1 < \lambda_2$, and k be a non-negative integer. $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY is in* P.

The theorem also implies the following.

Corollary 57. *Let F be a graph with degree values $\lambda_1 < \lambda_2$ and k be a non-negative integer. If F is such that all vertices of $V_{\lambda_1} \cup V_{\lambda_2}$ have the same number of neighbors in V_{λ_3} and $k < \lambda_3 + \lambda_1$, then one can solve $(\Delta \leq k)$-F-OVERLAY in polynomial time.*

Proof. Let j be the number of neighbors in V_{λ_3} that each vertex of $V_{\lambda_1} \cup V_{\lambda_2}$ has.

Let H be an $|F|$-uniform hypergraph. Observe that if a vertex has degree λ_3 in the subgraph induced by a hyperedge, then it cannot be in any other hyperedge, for otherwise it would have degree at least $\lambda_3 + \lambda_1 > k$. Therefore, we first check whether every hyperedge S contains a set $I(S)$ of $|V_{\lambda_3}|$ vertices that are uniquely in S. If not, then we return 'No'. If it is the case, let $F' = F - V_{\lambda_3}$ and H' be the hypergraph whose vertex set is $V(H) \setminus \bigcup_{S \in E(H)} I(S)$ and whose hyperedge set is $\{S \setminus I(S) \mid S \in E(H)\}$, and let $k'(v) = k - j \cdot d_H(v)$ for every $v \in V(H')$. Since all vertices of $V_{\lambda_1} \cup V_{\lambda_2}$ have exactly j neighbors in V_{λ_3}, it is then easy to prove the following claim:

Claim 57.1. *There is an (F,H,k)-graph G if and only if there is a graph G' F'-overlaying H' such that $d_G(v) \leq k'(v)$ for all $v \in V(H')$.*

Deciding whether such a graph G' exists can be done similarly as the proof of Theorem 55 replacing k by $k'(v)$. $\square$

The proofs of Theorem 55 and Corollary 57 directly generalize to $(\mathbf{d},k)$-SPLITTING for general hypergraphs.

Theorem 58. *Let $\mathbf{d}$ be a non-decreasing sequence and a positive integer k.*

(i) *If* $\mathbf{d}$ *contains at most two different values, then* $(\mathbf{d}, k)$-SPLITTING *is polynomial-time solvable.*

(ii) *If* $\mathbf{d}$ *is a non-decreasing sequence with three values* $\lambda_1 < \lambda_2 < \lambda_3$*, then* $(\mathbf{d}, k)$-SPLITTING *is polynomial-time solvable for* $k < \lambda_3 + \lambda_1$.

4.4.3 NP-completeness for graphs with $\lambda_1 = 1$

The previous section showed that $(\mathbf{d}, k)$-SPLITTING and NEAT $(\Delta \le k)$-F-OVERLAY is in P for $\mathbf{d}$ with at most two different values. It remains the case for $\mathbf{d}$ with at least three values. In this section, we aim to prove the following.

Theorem 59. *If* $\mathbf{d}$ *is a sequence with at least three values and with minimum value* 1*, then* $\mathrm{np}(\mathbf{d}) \le \mathrm{np}'(\mathbf{d}) \le +\infty$.

We first restrict an arbitrary sequence $\mathbf{d}$ to only one with only three different values.

Lemma 60. *Let* k *be an integer, let* $\mathbf{d} = \{d_1, d_2, \dots, d_p\}$ *be a non-decreasing sequence such that* $d_p \le k$*, and let* $\mathbf{d}' = \{d_1, d_2, \dots, d_{p'}\}$ *be a subsequence for some* $p' \le p$.

(i) *If* $(\mathbf{d}', k)$-SPLITTING *is* NP*-complete, then* $(\mathbf{d}, k)$-SPLITTING *is* NP*-complete.*

(ii) *If* NEAT $(\mathbf{d}', k)$-SPLITTING *is* NP*-complete, then* NEAT $(\mathbf{d}, k)$-SPLITTING *is* NP*-complete.*

Proof. Let H' be a p'-uniform hypergraph with hyperedges $S'_1, \dots, S'_m$. Let H be the p-uniform hypergraph obtained from H' as follows: for each $j \in [m]$, we add a set $X_j = \{x^j_i \mid i \in [p - p']\}$ of $p - p'$ new vertices and define $S_j = S'_j \cup X_j$.

Assume that there is a $(\mathbf{d}', k)$-splitting $\boldsymbol{\partial}'$ of H'. Let $\boldsymbol{\partial}$ be defined as follows. For every $j \in [m]$, let $\partial_{S_j}(v) = \partial'_{S'_j}(v)$ if $v \in S'_j$ and $\partial_{S_j}(x^j_i) = d_{p'+i}$ for all $i \in [p - p']$. One easily checks that $\boldsymbol{\partial}$ is a $(\mathbf{d}, k)$-splitting of H.

Reciprocally, assume that there is a $(\mathbf{d}, k)$-splitting $\boldsymbol{\partial}$ of H. An *inversion* of $\boldsymbol{\partial}$ is a pair of vertices (v, x) with $v \in S_j$ and $x \in X_j$ for some j such that $\partial_{S_j}(v) > \partial_{S_j}(x)$. Without loss of generality, we may assume that $\boldsymbol{\partial}$ is the $(\mathbf{d}, k)$-splitting of H with the minimum number of inversions. Then $\boldsymbol{\partial}$ has no inversion (v, x) for otherwise replacing ∂_{S_j} by the bijection obtained from it by swapping the images of v and x yields a $(\mathbf{d}, k)$-splitting of H with less inversions. Henceforth for each $j \in [m]$ the restriction $\partial'_{S'_j}$ of ∂_{S_j} to S'_j is a bijection from S'_j into $\mathbf{d}'$. By construction $\boldsymbol{\partial}' = (\partial'_{S'_j})_{j \in [m]}$ is a $(\mathbf{d}', k)$-splitting of H'.

Thus there is a $(\mathbf{d}, k)$-splitting $\boldsymbol{\partial}$ of H if and only if there is a $(\mathbf{d}', k)$-splitting $\boldsymbol{\partial}'$ of H'. This proves (i).

Observe that if H' is neat then H is also neat. This the above reduction also yields (ii). $\square$

It suffices now to prove for $\mathbf{d}$ with three values. Let F be a graph with the degree sequence $\mathbf{d}$ which has three degree values $1 = \lambda_1 < \lambda_2 < \lambda_3$. We will prove the NP-completeness of NEAT $(\Delta \leq k)$-F-OVERLAY Hence, it implies for NEAT $(\mathbf{d}, k)$-SPLITTING.

Theorem 61. *Let* $\mathbf{d} = (d_1, \ldots, d_p)$ *be a non-decreasing sequence with exactly three values* $1 = \lambda_1 < \lambda_2 < \lambda_3$. *Then* $\mathrm{np}'(\mathbf{d}) \leq \max\{\lambda_2 + \lambda_3, d_2 + 2\lambda_2\}$.

Proof. Let $k \geq \max\{\lambda_2 + \lambda_3, d_2 + 2\lambda_2\}$. Set $\delta = k - \lambda_2 - \lambda_3 + 1$, $\beta = k - \lambda_3 + 1$, $\gamma = k - \lambda_2 + 1$, and $\theta = \max\{0, k - 2\lambda_3 + 1\}$.

We give a reduction from 3-COLORING on 4-regular graphs to $(\mathbf{d}, k)$-SPLITTING. Given a graph 4-regular G, we construct a p-uniform hypergraph H as follows.

- For each vertex $v \in V(G)$, create a *vertex gadget* H_v as follows.
  - We first create a hyperedge $S_v = \{c_v^1, c_v^2, c_v^3\} \cup X_v \cup Y_v$ with $|Y_v| = \alpha_3 - 1$ and $|X_v| = p - \alpha_3 - 2$, and then add β hyperedges pendant at each vertex $x_v \in X_v$.
  - We create a binary tree T_v^i with vertex set $\{c_v^i, a_v^i, b_v^i, \ell_v^{i,1}, \ell_v^{i,2}, \ell_v^{i,3}, \ell_v^{i,4}\}$ for each $i \in [3]$, and edge set $\{c_v^i a_v^i, c_v^i b_v^i, a_v^i \ell_v^{i,1}, a_v^i \ell_v^{i,2}, b_v^i \ell_v^{i,3}, b_v^i \ell_v^{i,4}\}$. In this tree, c_v^i is the parent of a_v^i and b_v^i, a_v^i is the parent of $\ell_v^{i,1}$ and $\ell_v^{i,2}$, and b_v^i is the parent of $\ell_v^{i,3}$ and $\ell_v^{i,4}$.
  - For each edge $e = xy$ of T_v^i with x the parent of y in T_v^i, we construct an *edge-gadget* containing x, y, a new vertex z_e, and four disjoint sets $U_e^1, W_e^1, U_e^2, W_e^2$ of new vertices, U_e^1 of size $\alpha_1 - 1$, W_e^1 of size $p - \alpha_1 - 1$, U_e^2 of size $p - \alpha_3 - 1$, W_e^2 of size $\alpha_3 - 1$. We add the hyperedges $S_e^1 = \{x, z_e\} \cup U_e^1 \cup W_e^1$ and $S_e^2 = \{z_e, y\} \cup U_e^2 \cup W_e^2$. We add δ hyperedges pendant at x, γ pendant hyperedges at each vertex of U_e^1 and β hyperedges pendant at each of U_e^2, δ hyperedges pendant at z_e.

  For convenience, for any leaf $\ell_v^{i,j}$, we denote by $S_v^{i,j}$ the hyperedge containing the vertex $\ell_v^{i,j}$.
- For every vertex $v \in V(G)$, let e_v^1, e_v^2, e_v^3, e^4, be an ordering of the edges incident to v. For each edge $vw \in E(G)$, let j_v and j_w be the indices such that $vw = e_v^{j_v} = e_w^{j_w}$. Then, for all $i \in [3]$, we identify the vertices ℓ_v^{i,j_v} and ℓ_w^{i,j_w}, and at such a vertex, we add θ pendant hyperedges.

Let us prove that there is a 3-coloring of G if and only if there is a $(\mathbf{d}, k)$-splitting $\boldsymbol{\partial}$ of H.

Assume first that there is a $(\mathbf{d}, k)$-splitting $\boldsymbol{\partial}$ of H. Observe that for any $v \in V(G)$, no vertex in X_v can be λ_3-vertex since there are β pendant hyperedges at such a vertex in which it has value at least 1. Therefore, at least one vertex of $\{c_v^1, c_v^2, c_v^3\}$ is a λ_3-vertex in S_v. Moreover, since the vertices of Y_v are contained in exactly one hyperedge, we may assume that all vertices of Y_v are λ_3-vertices. Hence, exactly one vertex of $\{c_v^1, c_v^2, c_v^3\}$ is a λ_3-vertex in S_v.

Let ϕ be the 3-coloring of G defined by $\phi(v) = i$ if c_v^i is a λ_3-vertex of S_v. We will prove that ϕ is proper. We need the following claim.

We have the following claim.

Claim 61.1. *Let $v \in V(G)$ and $i \in \{1,2,3\}$ .If c_v^i is a λ_3-vertex of S_v, then $\ell_v^{i,j}$ is a λ_3-vertex of $S_v^{i,j}$ for all $j \in [4]$.*

Proof of claim: It suffices to prove that if $e = xy$ is an edge of T_v^i with x the parent of y in T_v^i and x is a λ_3-vertex of its *parent hyperedge* (that is S_v if $x = c_v^i$ or S_{wx}^2 with w the parent of x in T_v^i) then y is a λ_3-vertex of S_e^2.

Assume that x is a λ_3-vertex of its parent hyperedge. Then x must be a 1-vertex in S_e^1 for otherwise $\Sigma_{\boldsymbol{\partial}}(v)$ would be greater than k. Moreover any $u \in U_e^1$ must be a 1-vertex in S_e^1 because it is in γ pendant hyperedges. Therefore z_e is not λ_1-vertex and so has value at least λ_2 in S_e^1. Now because there are pendant hype δ pendant hyperedges at z_e, this vertex is not a λ_3-vertex of S_e^2. Moreover no vertex of U_e^2 is a λ_3-vertex of S_e^2 because each of them is in β pendant hyperedges. Therefore y must be a λ_3-vertex of S_e^2. ◁

Consider now an edge vw in $E(G)$ and $i \in \{1,2,3\}$. By construction the vertices ℓ_v^{i,j_v} and ℓ_w^{i,j_w} are identified. This vertex is in θ pendant hyperedges, so it cannot be λ_3-vertex in both S_v^{i,j_v}, S_w^{i,j_w}. Thus, by Claim 61.1, either c_v^i is not a λ_3-vertex of S_v or c_w^i is not a λ_t-vertex of S_w. Hence, there is no $i \in [3]$ such that c_v^i is a λ_3-vertex of S_v and c_w^i is an λ_3-vertex of S_w. In other words, $\phi(v) \neq \phi(w)$.

Therefore the 3-coloring is proper, and so G is 3-colorable.

Conversely, assume that G has a proper 3-coloring ϕ. Let us define $\boldsymbol{\partial} = \{\partial_S) \mid S \in E(H)\}$ by doing the following for every $v \in V(G)$ with $i = \phi(v)$ and $J = \{j_1, j_2\} = [3] \setminus \{i\}$.

- Pick a bijection ∂_{S_v} from S_v into $\mathbf{d}$ such that $\partial_{S_v}(c_v^i) = \lambda_3$, $\partial_{S_v}(c_v^{j_1}) = d_1$, $\partial_{S_v}(c_v^{j_2}) = d_2$, and $\partial_{S_v}(u) = \lambda_3$ for all $u \in Y_v$.
- For each edge $e = xy$ of T_v^i with x parent of y, pick a bijection $\partial_{S_e^1}$ from S_e^1 into $\mathbf{d}$ such that $\partial_{S_e^1}(u) = 1$ for all $u \in U_e^1 \cup \{x\}$ and $\partial_{S_e^1}(z_e) = \lambda_2$, and pick a bijection $\partial_{S_e^2}$ from S_e^2 into $\mathbf{d}$ such that $\partial_{S_e^2}(z_e) = 1$ and $\partial_{S_e^2}(u) = \lambda_3$ for every $u \in W_e^2 \cup \{y\}$.
- For each $j \in J$ and each edge $e = xy$ of T_v^j with x parent of y, pick a bijection $\partial_{S_e^1}$ from S_e^1 into $\mathbf{d}$ such that $\partial_{S_e^1}(u) = 1$ for all $u \in U_e^1 \cup \{z_e\}$ and $\partial_{S_e^1}(x) = \lambda_2$, and pick a bijection $\partial_{S_e^2}$ from S_e^2 into $\mathbf{d}$ such that $\partial_{S_e^2}(u) = \lambda_3$ for every $u \in W_e^2 \cup \{z_e\}$ and $\partial_{S_e^2}(y) = \lambda_1$.
- Finally for each hyperedge S pendant at a vertex x, we pick a bijection ∂_S from S into $\mathbf{d}$ such that $\partial_S(x) = \lambda_1$.

Let us check that $\boldsymbol{\partial}$ is a $(\mathbf{d}, k)$-splitting of H.

Let u be a vertex of H. We need to check that $\Sigma_{\boldsymbol{\partial}}(u) \leq k$.

Assume now that u is in two vertex-gadgets H_v and H_w. By construction, vw is an edge of G and there is i such that $u = \ell_v^{i,j_v} = \ell_w^{i,j_w}$. Moreover, since ϕ is a proper 3-coloring of G, either $\phi(v) \neq i$ or $\phi(w) \neq i$. Thus by definition of $\boldsymbol{\partial}$, u is a 1-vertex in one of S_v^{i,j_v}, S_w^{i,j_w}. Therefore, $\Sigma_{\boldsymbol{\partial}}(u) = \partial_{S_v^{i,j_v}}(u) + \partial_{S_w^{i,j_w}}(u) + \theta \leq 1 + \lambda_3 + \theta \leq k$.

Henceforth we may assume that u is a unique vertex gadget H_v for some $v \in V(G)$.

If u is in a unique hyperedge S, then $\Sigma_{\boldsymbol{\partial}}(u) = \partial_S(u) \leq \lambda_3 \leq k$. This is in particular the case if u is in a hyperedge pendant at a vertex distinct from u, or if $u \in Y_v$, or if $u \in W_e^1 \cup W_e^2$ for an edge e of T_v^i for some $i \in [3]$.

If $u \in X_v$ or U_e^2 for some edge e of some T_v^i, then $\Sigma_{\boldsymbol{\partial}}(u) \leq \beta + \lambda_2 \leq k$.

If $u \in U_e^1$ for some edge e of some T_v^i, then $\Sigma_{\boldsymbol{\partial}}(u) \leq \gamma + 1 \leq k$.

Suppose $u = c_v^i$ for $i \in [3]$. If $i = \phi(v)$, then u is a λ_3-vertex in S_v and a 1-vertex in both $S^1_{c_v^i a_v^i}$ and $S^1_{c_v^i b_v^i}$, so $\Sigma_{\boldsymbol{\partial}}(u) = \lambda_3 + 2 \leq \lambda_3 + \lambda_2 \leq k$. If $i \neq \phi(v)$, then $\partial_{S_v}(u) \in \{d_1, d_2\}$, and u is a λ_2-vertex in both $S^1_{c_v^i a_v^i}$ and $S^1_{c_v^i b_v^i}$, so $\Sigma_{\boldsymbol{\partial}}(u) \leq d_2 + 2\lambda_2 \leq k$.

Assume $u = a_v^i$ or $u = b_v^i$. If $\phi(v) = i$, then u is a λ_3-vertex in its parent hyperedge and a 1-vertex in the two other hyperedges containing it. Thus $\Sigma_{\boldsymbol{\partial}}(u) = \lambda_3 + 2 \leq k$. If $\phi(v) \neq i$, then u is a 1-vertex in its parent hyperedge and a λ_2-vertex in the two other hyperedges containing it. Thus $\Sigma_{\boldsymbol{\partial}}(u) = 1 + 2\lambda_2 \leq k$. $\square$

Theorem 61 and Lemma 60 immediately imply Theorem 59.

Conclusion of Part I

The first part was to study the complexity dichotomy of Overlaying problems which we aim to answer the general following problem.

Problem 62. Characterize the pairs $(\mathcal{F}, k)$ for which MAX $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY or $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY is polynomial-time solvable and those for which it is NP-complete.

A first step to answer this problem is to address it with a graph F. We gave the complete dichotomy of both problems for $F = O_p$ the graph on p vertices with one edge. Theorem 24 characterizes the complexity of MAX $(\Delta \leq k)$-F-OVERLAY which left us to study on $(\Delta \leq k)$-F-OVERLAY. It is more complicated for $(\Delta \leq k)$-F-OVERLAY, some evidences are given in Section 4.1 which presents several polynomial cases for $k \geq \Delta(F)$. In order to attack the problem, it would be helpful to prove the following conjecture.

Conjecture 63. If $(\Delta \leq k)$-F-OVERLAY is $\mathcal{NP}$-complete, then $(\Delta \leq k{+}1)$-F-OVERLAY is also $\mathcal{NP}$-complete.

We gave several evidences to this conjecture. We first showed that the NP-completeness for k implies one for $k + \delta(F)$. In particular, the conjecture holds for $\delta(F) = 1$.

We devoted a partial answer to Problem 62 for $(\Delta \leq k)$-F-OVERLAY which is the ultimate NP-completeness of the problem for every graph F different from K_p and $\overline{K}_p$ in Section 4.2. Precisely, we proved that $\mathrm{np}(F) < +\infty$ if and only if F is **standard**, that is neither a complete graph nor an anticomplete graph. It would be nice to get the characterization asked by this problem. Such a characterization might be complicated and hard to obtain, but we can certainly obtain further results. Indeed, in our proof, we made no attempt to minimize the upper bound on $\mathrm{np}(F)$, but just to prove such a bound exists. In fact, our proof shows the general upper bound $\mathrm{np}(F) \leq 8|F|\delta(F)$ for every standard graph F. However, there are many graphs for which the proof shows

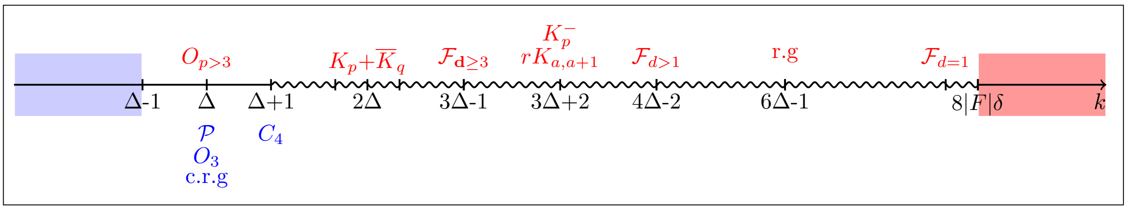


Figure 4.9: **The complexity dichotomy of $(\Delta \leq k)$-F-Overlay for F standard.** (see below Table 4.1 for the definitions of the notations) P and NP-complete results are respectively corresponding to pairs (F, k) and (F, k). For $k < \Delta - 1$, the problem is in P (red box) but is it NP-complete for $k \geq 8|F|\delta$ (blue box). For a pair (F, k), k is an upper bound of $\mathrm{np}(F)$.

Δ, δ	$\Delta(F), \delta(F)$ the maximum and minimum degree of F
$O_{p>3}$	the graph on p vertices and one edge with $p > 3$
(c.)r.g	(connected) regular graphs
$rK_{a,a+1}$	disjoint union of r copies of $K_{a,a+1}$
$K_p + \overline{K}_q$	disjoint union of K_p and $\overline{K}_q$
$\mathcal{F}_{\mathbf{d}\geq 3}$	the family of graphs with at least three degree values
K_p^-	K_p minus one edge
$\mathcal{F}_{d>1}$	the family of graphs F with $\mathbf{d}$ such that $\exists i$, $d_i - d_{i-1} > 1$
$\mathcal{F}_{d=1}$	the family of graphs F with $\mathbf{d}$ such that $\forall i$, $d_i - d_{i-1} = 1$

Table 4.1: **Table of notations in Figure 4.9.**

$\mathrm{np}(F) \leq 6|F|$ or even $\mathrm{np}(F) \leq 6\Delta(F)$ (see Figure 4.9 for a description of results). It motivates the following question.

Problem 64. Does there exist a constant c such that $\mathrm{np}(F) \leq c \cdot \Delta(F)$ for every standard graph F ?

Getting lower bounds would also be interesting. The trivial lower bound is $\mathrm{np}(F) \geq \Delta(F)$. There are graphs for which this lower bound is attained, the graphs with one edge of at least 4 vertices for example. It would be nice to characterize the graphs corresponding to the difference $\mathrm{np}(F) - \Delta(F)$. The largest known value is 2 for C_4, the cycle on four vertices by Theorem 29.

The results for $\mathcal{F} = \{F\}$ of size 1 are useful because the NP-completeness of the problems for F implies the NP-completeness for the families $\mathcal{F}$ such that $F \in \mathcal{F}$ and $|F| \neq |F'|$ for all other $F' \in \mathcal{F}$. It would be nice to answer whether we can also extend the ultimate NP-completeness of a standard graph F to a family containing F.

Conjecture 65. If $\mathcal{F}$ contains a standard graph F and both $K_{|F|}, \overline{K}_{|F|}$ are not in $\mathcal{F}$, then $\mathrm{np}(\mathcal{F}) < +\infty$.

This is far from trivial for many graph families. For example, depending on the degree sequences of graphs in a family $\mathcal{F}$. Let $F, F' \in \mathcal{F}$ whose degree sequences respectively $\mathbf{d}, \mathbf{d}'$ such that $|F| = |F'|$ and $d_i \leq d'_i$ for all $i \in [|F|]$, then it be easier to use only F overtime to overlay a hyperedge rather than F'. Otherwise, it is difficult to decide whether a hyperedge should be F-overlaid or F'-overlaid.

Since the NP-completeness results for many family $\mathcal{F}$ are obtained from the NP-completeness of a graph $F \in \mathcal{F}$, an interesting question is to ask for polynomiality results associated to $k \geq \max_{F \in \mathcal{F}} \Delta(F)$.

Problem 66. Let $\mathcal{F}$ be a family such that $(\Delta \leq k_F)$-F-OVERLAY is in P for $F \in \mathcal{F}$, is $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY in P for some $\max_{F \in \mathcal{F}} \Delta(F) \leq k \leq \max_{F \in \mathcal{F}} k_F$?

For $\mathcal{F}$ a family of graphs with the same degree sequence such that it has at most two degree values, NEAT $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY is in P by Corollary 56. Now, let $\mathcal{F} = \{K_2, C_4\}$ and $k = 2$, we know that $(\Delta \leq 2)$-F-OVERLAY is in P for any F in $\mathcal{F}$. For a hypergraph H in which its hyperedges are of size either 2 or 4, it is easy to overlay all hyperedges of size 2 (by taking the edge between the two vertices of a hyperedge). Thus, these edges are sure in G and we have to complete G using these edges. It motives the following question.

Problem 67. let H be a hypergraph and a graph G (which overlays a subset of hyperedges of H), can G extend to an $(\mathcal{F}, H, k)$-graph?

Let H be a hypergraph and a graph G (which overlays a subset of hyperedges of H), can G extend to an $(\mathcal{F}, H, k)$-graph?

Furthermore, by the set of given edges of G, we can compute the remaining degree of a vertex which would be used to extend G to a graph $\mathcal{F}$-overlaying H (for an example with $\mathcal{F}$ above, after overlaying all hyperedges of size 2, this leaves hyperedges of size 4 and the remaining degree of any vertex is either 0, 1 or 2 (it is when this vertex respectively, belongs to 2, 1 or no hyperedges of size 2). Thus, it leads to the following problem.

Problem 68. Given a hypergraph H and a function $f : V(H) \to [k]$, does it exist a graph G F-overlaying H such that $d_G(v) \leq f(v)$?

In the end, we have studied the restriction of $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY to neat hypergraphs NEAT $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY and $(\mathbf{d}, k)$-SPLITTING. Its restriction on neat hypergraphs is NEAT $(\mathbf{d}, k)$-SPLITTING is equivalent to NEAT $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY for $\mathcal{F}$ the graph family with degree sequence $\mathbf{d}$. We conjecture the following.

Conjecture 69. $(\mathbf{d}, k)$-SPLITTING is NP-complete if and only if NEAT $(\mathbf{d}, k)$-SPLITTING is NP-complete.

We have seen NEAT $(\Delta \leq k)$-$\mathcal{F}$-OVERLAY and $(\mathbf{d}, k)$-SPLITTING are in P for $\mathbf{d}$ having at most two values; or $\mathbf{d}$ with three values and $k < \lambda_3 + \lambda_1$. For $\mathbf{d}$ with at least three values and if $\lambda_1 = 1$, then they are ultimately NP-complete which is precisely

$\mathrm{np}'(\mathbf{d}) < \max\{3\lambda_3, \lambda_t\}$ by Theorem 61. It would be nice to relax the condition $\lambda_1 = 1$ to an arbitrary value.

Conjecture 70. If $\mathbf{d}$ be a sequence of positive integers with at least three different values, then $\mathrm{np}(\mathbf{d}) = \mathrm{np}'(\mathbf{d}) < +\infty$.

By Lemma 60, it suffices to prove the conjecture for sequences with only three values and the greatest one appearing only once.

Determining the complexity is a first step of studying problems. We know how to solve those in P. For an NP-complete problem, it would be interesting to design algorithms solving the problem. As a start, we can build heuristic algorithms or a simple greedy algorithm which can be helpful to compute an approximation-ratio to study on approximation algorithms.

For each pair (F, k) such that MAX $(\Delta \leq k)$-F-OVERLAY is NP-complete and $(\Delta \leq k)$-F-OVERLAY is polynomial-time solvable, we can consider the parameterized complexity of MAX $(\Delta \leq k)$-F-OVERLAY when parameterized by $|E(H)| - s$.

Furthermore, these problems are motivated by biology problems, it is necessary to attack on problems with instances restricted to biological properties (*e.g.* an upper bound on the maximum degree k, typical numbers of subunits per complex).

Part II

Conflict Coloring problems

Chapter 5

Introduction

5.1 Determining the high resolution model with DOMINO

Assume that the low resolution model of a molecular assembly (MA) provides the interaction graph of its subunits. We are given a finite set of conformations for each subunits. A scoring function measures the quality of the contact between two conformations of a subunit-contact in the assembly. An issue of determining the high resolution model for a MA is to chose a conformation for each subunit in which together, they are well structured in the whole MA. This problem was defined using DOMINO [LTSW09] which outputs a set of conformations for subunits maximizing a global function. This can be modeled as a coloring problem as the following.

Let $G = (V, E)$ be an interaction graph representing a given assembly in which V represents the set of subunits and E is the set of contacts between them. We label the set of conformations for each subunit to a color set of the corresponding vertex in V. We assume that every vertex has the same number of colors which is an integer k, hence the set of colors is $[k] = \{1, 2, \dots, k\}$. For every $uv \in E$, the scoring function provides the edge-weighted complete bipartite graph K^k_{uv} which is defined as follows: $V(K^k_{uv}) = \{(u, i), i \in [k]\} \cup \{(v, j), j \in [k]\}$, $E(K^k_{uv}) = \{(u, i)(v, j), i, j \in [k]\}$, and the weight function $w^k_{uv} : E(K^k_{uv}) \to \mathbb{R} \cup \{-\infty\}$. We denote by $w^k = \{w^k_{uv}, uv \in E\}$ the set of weight functions over all the edges $uv \in E$.

Hence, for every vertex $v \in V$, the vertex set $\{(v, i), i \in [k]\}$ represents the k possible conformations of the subunit associated with v. For every $uv \in E$, the weight $w^k_{uv}((u, i)(v, j))$ represents the *score* between the i-conformation of u and the j-one of v. Meaning by the scoring function, the higher the score, the better the connection between the two associated subunits.

In general, the number of colors is not the same for all vertices. However, we can easily change it so that all vertices have the same number k of colors: it suffices to add extra "fake" colors and to assign weight $-\infty$ to all edges incident with these colors in the graph K^k_{uv}. It explains our assumption of the same number of colors for all vertices.

We are now able to formalize DOMINO as a graph coloring problem, which consists in finding a coloring such that the sum of the weights is at least a given lower-bound.

DOMINO.
Input: A graph $G = (V, E)$, an integer $k \geq 1$, a set of weight functions w^k, and $\gamma \in \mathbb{R}$.
Question: Does there exist a coloring c of G such that $\sum_{uv \in E} w^k_{uv}((u, c(u))(v, c(v))) \geq \gamma$?

In structural biology, conformations of subunits are compatible according to the lock-and-key model. Thus, the total score of a set of conformations reflects whether the assembly defined by this set is well constructed and robust. It is the property used to define DOMINO.

DOMINO is clearly in NP. We can easily construct a reduction from k-coloring, as follows. Given a graph G, we define the weight function by taking, for any $uv \in E(G)$, $w^k_{uv}((u, i)(v, i)) = 0$ and $w^k_{uv}((u, i)(v, j)) = 1$ for $j \neq i$. Then G is k-colorable if and only if it has a coloring c with total weight at least $\gamma = |E(G)|$.

DOMINO with additional weights on the vertices, has been mostly studied from biology point of view and some works are related to computational approaches. As a corollary of NP-completeness of DOMINO, this closely related problem is also NP-complete, which has been proven in [Lat94]. There are also methods to solve this latter problem with heuristic approaches, using tree decomposition [LTSW09, XJB05] or linear programming [XLKX03].

5.2 Conflict coloring

An important constraint, from computational structural biology, consists in having a lower bound $\lambda \in \mathbb{R}$ on the weight of every edge $uv \in E$. More precisely, the coloring c must satisfy $w^k_{uv}((u, c(u)), (v, c(v))) \geq \lambda$ for every $uv \in E$. This constraint may imply a stable structure of the macromolecular assembly, that is without "bad connection" between two subunits.

To take into account such a constraint, we define a *conflict k-graph* M^λ_{uv} for every edge $uv \in E$ as follows. M^λ_{uv} is the subgraph of K^k_{uv} such that $(u, c(u))(v, c(v)) \in E(M^\lambda_{uv})$ if and only if $w^k_{uv}((u, c(u)), (v, c(v))) < \lambda$. We denote by $\mathcal{M}^\lambda = \{M^\lambda_e$ for all $e \in E\}$, the *set of conflict k-graphs* (k-SoC). An edge $uv \in E$ is *$\mathcal{M}^\lambda$-conflicting* for a k-coloring c if $(u, c(u))(v, c(v)) \in E(M^\lambda_{uv})$, otherwise it is *$\mathcal{M}^\lambda$-fulfilled*.

It leads to the following problem with $\mathcal{M} = \mathcal{M}^\lambda$.

FULFILL COLORING.
Input: A graph $G = (V, E)$, an integer k, a k-SoC $\mathcal{M}$, and a positive integer q.
Question: Does there exist a k-coloring c of G with at least q $\mathcal{M}$-fulfilled edges ?

An important particular case of this problem is when $q = |E|$. In that case, the problem, called CONFLICT COLORING, is to determine whether G is *$\mathcal{M}$-conflict colorable*, that is whether it admits a k-coloring with no $\mathcal{M}$-conflict edges. Such a coloring is called an *$\mathcal{M}$-conflict coloring*. This notion of conflict coloring was introduced by Dvořák and Postle [DP17] and Fraigniaud et al. [FHK16].

We shall consider the restrictions of FULFILL COLORING, and CONFLICT COLORING to the case when k is fixed. These are called FULFILL k-COLORING, and CONFLICT k-COLORING, respectively.

Observe that FULFILL COLORING is the particular case of DOMINO for which $\gamma = q$ and $w^k((u, c(u)(v, c(v))) = 1$ if $((u, c(u)(v, c(v)) \notin E(\mathcal{M}_{uv})$ and $w^k((u, c(u)(v, c(v))) = 0$ otherwise.

We finally define variants of FULFILL COLORING and its restrictions, in which $\mathcal{M}$ is contained in some restricted family $\mathcal{B}$ of bipartite graphs. For example, $\mathcal{B}$-CONFLICT k-COLORING is the restriction of CONFLICT k-COLORING in which each conflict k-graph M_e must belong to $\mathcal{B}$. We denote by $\mathcal{B}(n, m)$ the class of subgraphs of $K_{n,n}$ with m edges. Figure 5.1 describes an instance of $\mathcal{B}(3, 2)$-CONFLICT 3-COLORING and a solution. When $\mathcal{B}$ is the family of matchings (i.e. bipartite graphs with maximum degree 1), $\mathcal{B}$-CONFLICT k-COLORING is known as CORRESPONDENCE COLORING or DP-COLORING (formally defined in the next section). Since it is a generalization of the classical notions of coloring and list coloring, it has been extensively studied over the past five years (see e.g., [Ber19, DP17, KO19, RSYG19]).

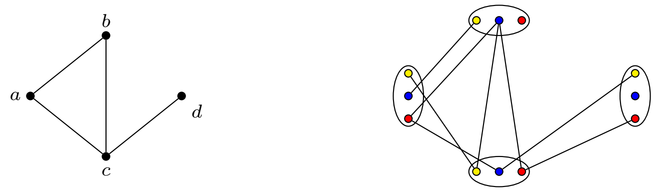


Figure 5.1: **Example of conflict coloring.** A graph G on four vertices (left) and a 3-SoC $\mathcal{M}$ of G (right) with 3 colors yellow, blue and red; $M_e \in \mathcal{B}(3, 2)$: for every $e \in E(G)$ contains two edges. An $\mathcal{M}$-Conflict Coloring c of G is $c(a) = c(c) = c(d) =$ "blue", $c(b) =$"red". Coloring c is a solution of $\mathcal{B}(3, 2)$-CONFLICT 3-COLORING.

5.3 Related problems

We have seen (PROPER) COLORING problem in Section 1.3.4. A generalization of this problem is LIST COLORING which is formally defined as follows: for a graph G, a list assignment L is a function from any vertex $v \in V(G)$ to $L(v)$ a list of colors; an L-coloring c of G is a proper coloring such that $c(v) \in L(v)$ for every $v \in V(G)$. The problem consists, given a graph G and a list assignment L to decide whether there is an L-coloring of G. A problem which is more general than LIST COLORING is CORRESPONDENCE COLORING (DP-COLORING), first defined in [DP17]. Given a graph G, a list assignment L and a function C from any edge $uv \in E(G)$ to a matching C_{uv} of the complete bipartite graph $K_{uv}^{L(u),L(v)}$ (the vertex set is $\{(u,i), i \in [L(u)]\} \cup \{(v,j), j \in [L(v)]\}$ and the edge set is $\{(u,i)(v,j), i \in [L(u)], j \in [L(v)]\}$. An (L,C)-coloring c of G is an L-coloring such that $(u,c(u))(v,c(v)) \notin E(C_{uv})$ for every edge $uv \in E(G)$. Hence, DP-COLORING generalizes LIST COLORING which is when C_{uv} contains only edges of the same colors in $L(u)$ and $L(v)$ for any $uv \in E(G)$. Furthermore DP-COLORING is a particular case of CONFLICT COLORING, since in this latter we can choose an arbitrary conflict graph rather than a matching.

Complexity dichotomies of some generalizations of coloring problems have been obtained. A typical example is H-COLORING.

5.3.1 H-COLORING and Constraint Satisfaction Problem (CSP)

Let H be a fixed graph (possibly directed and loops are allowed), whose vertices are *colors*. An *H-coloring* of a graph G is a coloring c assigning to each vertex of G a vertex of H such that $c(u)c(v) \in E(H)$ for any $uv \in E(G)$. Observe that k-COLORING is the case when H is the complete graph K_k. For a fixed (directed) graph H, H-COLORING problem is to decide whether a given (directed) graph G admits an H-coloring. A complexity dichotomy for such problems with H undirected has been known for several decades.

Theorem 71 ([HN90]). *Let H be an undirected graph. H-COLORING problem is polynomial-time solvable if H is bipartite or has a loop, and NP-complete otherwise.*

CSP is a very general problem which an instance is defined by a triple $(\mathcal{V}, \mathcal{D}, \mathcal{C})$ where $\mathcal{V} = \{V_1, \dots, V_n\}$ is a set of variables, $\mathcal{D} = \{D_1, \dots, D_n\}$ is a domain of values and $\mathcal{C} = \{C_1, \dots, C_m\}$ is a set of constraints. Each variable V_i can take a value in its domain D_i. Every constraint C_j is a triple (r_j, t_j, R_j) where r_j is a positive integer called *arity* of the constraint, t_j is an r_j-tuple of variables and R_j is an r_j-ary relation on t_j. An *evaluation* satisfies a constraint (r_j, t_j, R_j) if the variables of t_j are assigned values which satisfies R_j. An evaluation which satisfies all constraints is called a *solution*. Observe that H-COLORING and CONFLICT COLORING are particular cases of binary CSP where all constraints are binary relations (on variable domains corresponding to the vertex colors).

A finite constraint language Γ is a finite domain and a finite set of relations over this domain. CSP(Γ) is the constraint satisfaction problem where instances are only allowed to use constraints in Γ. This problem is equivalent to some H-COLORING for some directed graph in [FV98] by Feder and Vardi in 1998. They conjectured that CSP(Γ) is either in P or NP-complete (CSP conjecture). In 2017, this conjecture was independently proven by Zhuk [Zhu17] and Bulatov [Bul17] and yielded the following corollary for H-COLORING with directed graph H.

Corollary 72. *For a fixed directed graph H, H-COLORING is either in P or NP-complete.*

5.3.2 CONFLICT COLORING and comparisons to H-COLORING and CSP

For any graph H with vertex set $[k]$, we define the bipartite graph B_H as follows: the vertex set is $A \cup B$ in which A, B copies of $[k]$ and their vertices are respectively denoted by i_A, j_B for $i, j \in [k]$; if there is an (undirected) edge $ij \in E(H)$ (resp. directed $\overrightarrow{ij} \in E(H)$), then $i_A j_B, i_B j_A \in E(B_H)$ (resp. $i_A j_B \in E(B_H)$). Thus, the H-COLORING problem is equivalent to CONFLICT k-COLORING with $k = |V(H)|$ and the k-SoC containing only the complement $\overline{B_H}$ of B_H (*i.e.* $V(\overline{B_H}) = V(B_H)$ and $E(\overline{B_H}) = \{uv \mid \forall uv \notin E(B_H)\}$), see Figure 5.2. Hence, this problem and even its restriction $\mathcal{B}(k, |E(\overline{H})|)$-CONFLICT k-COLORING are more general than H-COLORING.

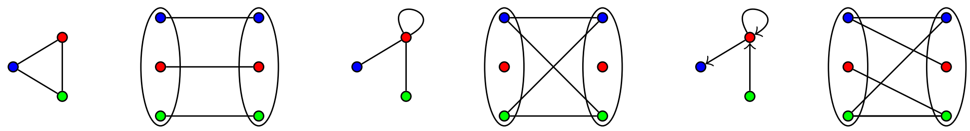


Figure 5.2: **Convert** H-COLORING **to** CONFLICT $|V(H)|$-COLORING. If $H_1 = K_3$, then the corresponding conflict graph is a subgraph of $K_{3,3}$ which contains three edges of same colors (two leftmost). The conflict graph associated to the second graph H_2 contains conflict (incompatible) edges which are complement to H. Note that first two graphs H are undirected, then their conflict graphs of an edge uv are symmetric but it is not true if H is directed, then it implicitly implies an orientation of edge uv to apply the conflict graph to uv. An example is shown in the last graph H_3 directed and its conflict graph.

A bipartite conflict graph M in a k-SoC $\mathcal{M}$ can be converted to a binary relation R_M with $(c, c') \in R_M \iff (c, c') \notin E(M)$ for all $c, c' \in [k]$. As a consequence we can reduce an instance of k-CONFLICT COLORING to the instance of CSP with variables $V(G)$ on domain $[k]$ and the set of constraints $\{R_{M_{uv}}, uv \in E(G)\}$ corresponding to the conflict edges. However, when $\mathcal{M}$ is restricted to a family $\mathcal{B}$ of bipartite graphs, the constraint language $\Gamma_{\mathcal{B}} = \{R_M, M \in \mathcal{B}\}$ gives a problem CSP($\Gamma_{\mathcal{B}}$) with more instances than $\mathcal{B}$-CONFLICT k-COLORING, in other words $\mathcal{B}$-CONFLICT k-COLORING is a restriction of CSP($\Gamma_{\mathcal{B}}$). Indeed, in CSP($\Gamma_{\mathcal{B}}$) we can choose arbitrary constraints among the relations and, in particular, multiple constraints on the same couple of variables; let say they are represented as conflict graphs $M_1, \ldots, M_\ell$. So this would correspond in $\mathcal{B}$-CONFLICT k-COLORING to a conflict graph for the same couple of vertices which is the *union* of $M_1, \dots, M_\ell$ (because CSP is a *conjunction* of constraints). From this reasoning we deduce the following:

Lemma 73. *If $\mathcal{B}$ is* closed by union, i.e. $(M, M' \in \mathcal{B} \implies (M \cup M') \in \mathcal{B})$, *then $\mathcal{B}$-*CONFLICT k-COLORING *is equivalent to* CSP($\Gamma_{\mathcal{B}}$) *with* $\Gamma_{\mathcal{B}} = \{R_M, M \in \mathcal{B}\}$.

As a consequence, complexity upper bounds on CSP($\Gamma_{\mathcal{B}}$) apply to $\mathcal{B}$-CONFLICT k-COLORING, and complexity lower bounds on $\mathcal{B}$-CONFLICT k-COLORING apply to CSP($\Gamma_{\mathcal{B}}$). In particular, when $\mathcal{B}$ is not closed by union, we cannot immediately apply the complexity lower bound results from CSP($\Gamma_{\mathcal{B}}$).

5.4 Our contributions

We first study the computational complexity of FULFILL-k-COLORING, CONFLICT-k-COLORING and their restrictions on graph families $\mathcal{B}(k, m)$ in Chapter 6.

If for every $uv \in E(G)$, the edge set of M_{uv} is the matching $\{\{(u, i), (v, i)\} \mid i \in [k]\}$, then the graph is $\mathcal{M}$-conflict colorable if and only if it is k-colorable. Therefore k-COLORING is a particular case of CONFLICT k-COLORING. It is well-known that k-COLORING is NP-complete if and only if $k \geq 3$ (Section 1.3.4). Henceforth, the problems CONFLICT k-COLORING and FULFILL k-COLORING are NP-complete for all $k \geq 3$. Thus, the problems CONFLICT COLORING and FULFILL COLORING are NP-complete. We prove in Theorem 75 that CONFLICT 2-COLORING is polynomial-time solvable. On the other hand, FULFILL 2-COLORING is more general than MAX CUT (decision version). This problem is given a graph and an integer, and aims to determine whether there is a partition of vertices into two disjoint sets such that there number of edges between them is at least the number. It is known to be NP-complete [Kar72]. Therefore FULFILL 2-COLORING is also NP-complete.

We next study $\mathcal{B}(k, m)$-FULFILL k-COLORING and $\mathcal{B}(k, m)$-CONFLICT k-COLORING. We prove polynomial vs. NP dichotomy results with respect to the considered values k and m. All those complexity results are summarized in Table 5.1.

Remark 74. Given a coloring, we can easily check whether it satisfies the problems FULFILL-COLORING or CONFLICT-COLORING in polynomial time. Thus, these two problems are in NP and we only prove the NP-hardness of NP-complete results.

In Chapter 7, we expose several algorithms solving the problems. As the decision problem MAX CUT is a particular case of FULFILL 2-COLORING, we investigate whether results for MAX CUT (for decision or optimization problem) can be also generalized to our problem on two colors. Hadlock [Had75] showed that MAX CUT is polynomial-time solvable for planar graphs. In contrast, we observe that FULFILL 2-COLORING is NP-complete (Remark 79). Goemans and Williamson [GW95] proved

	FULFILL k-COLORING	CONFLICT k-COLORING
$k = 1$	P	P
$k = 2$	NP-complete	P
$k \geq 3$	NP-complete	NP-complete
	$\mathcal{B}(k, m)$-FULFILL k-COLORING	$\mathcal{B}(k, m)$-CONFLICT k-COLORING
$m = 0$	P	P
$1 \leq m \leq k^2 - 3$	NP-complete	NP-complete
$m \in \{k^2 - 1, k^2 - 2\}$	NP-complete	P
$m = k^2$	P	P

Table 5.1: Dichotomy results on conflict coloring.

that MAX CUT is 0.87856-approximable using semidefinite programming. In Section 7.1, we use their approach to construct approximation algorithms for finding a coloring with maximum number of fulfilled edges (optimization version of FULFILL 2-COLORING).

Next, in Section 7.2, we show a polynomial-time algorithm that solves the maximization version of DOMINO for graphs with bounded treewidth. Similarly, we prove a polynomial-time algorithm that solves the problem (for bounded treewidth graphs) that consists in **determining the maximum threshold** λ for which, given a graph G, integers k, q, a set of weight functions w^k, there is a k-coloring of G that $\mathcal{M}^\lambda$-fulfills all edges. *For these studies on algorithms, we later know that there already exists an algorithm using tree decomposition for the optimization of* DOMINO **with additional weights on the vertices** *in computational biology communities. And we have mentioned the article in the end of Section 5.1.*

The last chapter describes an experimental works. We first introduce our setup to construct (geometric) instances of CONFLICT COLORING and then present some statistics which we obtain from our experiments. Namely, instances of CONFLICT COLORING are associated with *Voronoi diagrams* in the plane in which we retain as vertices those data points have a finite Voronoi region. Then, we build a k-SoC $\mathcal{M}$ by generating a set of k (randomly) perturbed polygons of the finite Voronoi polygon of a point, and given an intersection area threshold τ_c, a conflict between two perturbed polygons of adjacent Voronoi regions is when their intersection area is greater than τ_c (detail in Section 8.1). Hence, a random instance and a threshold τ_c yield a set of conflicts. The obtained statistics provide information on the relationship between the existence of a solution and thresholds which are derived from some local conditions.

Chapter 6

Complexity results

The aim of this chapter is to prove results in Table 5.1.

6.1 Unrestricted conflict coloring

We first establish the complexity of CONFLICT k-COLORING and FULFILL k-COLORING, and then consider the restrictions of the problems to the family $\mathcal{B}(k,m)$.

CONFLICT 1-COLORING and FULFILL 1-COLORING are trivial: there is only one possible color for each vertex, we just need to check which edges e are fulfilled, that is such that M_e is edgeless.

We now consider CONFLICT 2-COLORING and obtain the following.

Theorem 75. CONFLICT 2-COLORING *is polynomial-time solvable.*

Proof. Let $G = (V, E)$ be a graph and let M_{uv} be the conflict 2-graph for each edge $uv \in E$. If there is an edge $e \in E$ such that $M_e = K_{2,2}$, then the answer is 'No' because e cannot be fulfilled. If there is an edge $e \in E$ such that $M_e = K_{2,0}$ (edgeless graph), then e is always fulfilled. Without loss of generality, we assume that the conflict 2-graph of every edge is neither $K_{2,2}$ nor $K_{2,0}$.

Now, from G and the set of conflict 2-graphs M_{uv} for each edge $uv \in E$, we construct a 2-SAT formula Φ as follows.

1. For a vertex $v \in V$ and a color in $\{1, 2\}$, create a variable x_v for $(v, 1)$ and $\bar{x}_v$ for $(v, 2)$. So we denote by ℓ_v^i a literal of variable x_v, in which $\ell_v^1 = x_v$ and $\ell_v^2 = \bar{x}_v$.
2. For an edge $uv \in E$ with M_{uv}, we create a clause gadget $C_{uv} = \bigwedge_{(u,i)(v,j) \in E(M_{uv})} (\bar{\ell}_u^i \vee \bar{\ell}_v^j)$.

We will prove that G admits a conflict coloring c if and only if there is an assignment ϕ satisfying Φ.
Observe that, for any $uv \in E(G)$, the clause gadget C_{uv} contains a clause for each edge $(u,i)(v,j) \in M_{uv}$ which is the negation of $(\ell_u^i \wedge \ell_v^j)$. It implies that c does not conflict on uv by edge $(u,i)(v,j) \in M_{uv}$ if and only if $\neg(\ell_u^i \wedge \ell_v^j)$ is satisfiable by ϕ. Thus, uv is fulfilled by c if and only if C_{uv} is satisfiable by ϕ.
Therefore, one can determine c from ϕ by taking $c(v) = 1$ (resp. $c(v) = 2$) if $\phi(x_v) = true$ (resp. $\phi(x_v) = false$). □

In contrast to CONFLICT 2-COLORING, FULFILL 2-COLORING generalizes the well-known NP-complete problem MAX CUT, and thus it is also NP-complete. For $k \geq 3$, CONFLICT k-COLORING generalizes the NP-complete problem k-COLORING and so is NP-complete. Hence, FULFILL k-COLORING is also NP-complete.

Proposition 76. CONFLICT k-COLORING *is* NP-*complete for all* $k \geq 3$. FULFILL k-COLORING *is* NP-*complete for all* $k \geq 2$.

6.2 Restrictions to the family $\mathcal{B}(k, m)$

Now, we look at the restrictions of those problems when every conflict k-graph M_e is in $\mathcal{B}(k,m)$, that is a bipartite graph with k vertices in each part and m edges. We assume that $k \geq 2$. If $m = 0$ (resp. $m = k^2$), then the conflict k-graphs are edgeless (resp. complete bipartite) and so every k-coloring fulfills all edges (resp. no edges). Hence $\mathcal{B}(k,0)$-FULFILL k-COLORING and $\mathcal{B}(k,k^2)$-FULFILL k-COLORING are trivially polynomial-time solvable.

When $k = 2$, $\mathcal{B}(2,m)$-CONFLICT 2-COLORING is in P by Theorem 75. We prove the following for $\mathcal{B}(2,m)$-FULFILL 2-COLORING.

Theorem 77. $\mathcal{B}(2,m)$-FULFILL 2-COLORING *is* NP-*complete for* $m \in \{1,2,3\}$.

Proof. An easy observation is that $\mathcal{B}(2,2)$-FULFILL 2-COLORING is NP-complete since it generalizes MAX CUT. Thus, we now prove for graph classes $\mathcal{B}(2,1)$ and $\mathcal{B}(2,3)$.

1. $\mathcal{B}(2,1)$-FULFILL 2-COLORING is NP-hard.

A reduction from MAX-2-SAT problem which is known to be NP-hard [GJS76]. Given a 2-SAT formula Φ with N variables $x_1, \ldots, x_N$ and l clauses, each contains exactly 2 literals. We construct a graph G and the conflict 2-graph M_e for any $e \in E(G)$ as follows (see Figure 6.1 for an example):

- For each variable x_t, we create a vertex v_t, and if x_i appears in α_t clauses, then we create three copies $v_{t1}^r, v_{t2}^r, v_t^r$ for each r-occurrence of x_t, $r \in [\alpha_t]$. We add edges $v_{t1}^r v_t^r, v_{t2}^r v_t^r$ to $E(G)$; and $(v_{t1}^r, 1)(v_t^r, 2) \in M_{v_{t1}^r v_t^r}$ and $(v_{t2}^r, 2)(v_t^r, 1) \in M_{v_{t2}^r v_t^r}$. For $r \in [\alpha]$, we create a cycle on vertices v_t, v_{t1}^r, v_{t2}^r, and for (ordered

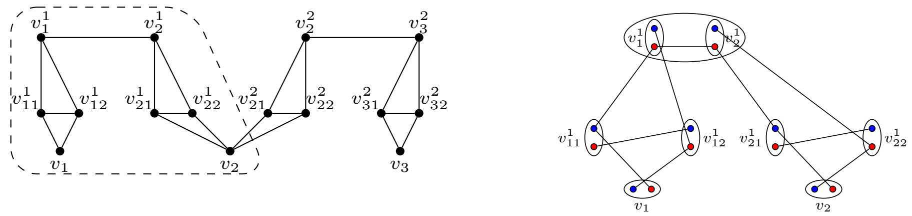


Figure 6.1: **An example of the construction for $\Phi = (x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3)$.** The graph G (left) and the corresponding conflict 2-graphs of edges in the dashed box (colors 1, 2 are respectively blue, red).

edge) $uv \in \{v_t v_{t1}^r, v_{t1}^r v_{t2}^r, v_{t2}^r v_t\}$, M_{uv} contains only edge $(u,2)(v,1)$. For conveniences, we call the subgraph on four vertices $v_t, v_{t1}^r, v_{t2}^r, v_t^r$ is a *v_t^r-graph* for every $r \in [\alpha_t]$.

- For each clause, assume that $C = (\ell_1 \vee \ell_2)$ where ℓ_1, ℓ_2 are respectively the r_1-th occurrence of x_p, r_2-th occurrence of x_q in C. We add to $E(G)$ the clause edge $e = v_p^{r_1} v_q^{r_2}$, and create its conflict 2-graph M_e as follows:
  - If $\ell_1 = x_p$ and $\ell_2 = x_q$, then $(v_p^{r_1}, 2)(v_q^{r_2}, 2) \in M_e$.
  - If $\ell_1 = x_p$ and $\ell_2 = \bar{x}_q$, then $(v_p^{r_1}, 2)(v_q^{r_2}, 1) \in M_e$.
  - If $\ell_1 = \bar{x}_p$ and $\ell_2 = x_q$, then $(v_p^{r_1}, 1)(v_q^{r_2}, 2) \in M_e$.
  - If $\ell_1 = \bar{x}_p$ and $\ell_2 = \bar{x}_q$, then $(v_p^{r_1}, 1)(v_q^{r_2}, 1) \in M_e$.

We shall prove that Φ admits an assignment which satisfies at least k clauses if and only if there is a 2-coloring c which fulfills at least $10l + k$ edges of G.

Assume that we have an assignment ϕ satisfying at least k clauses of Φ. Then, we take a coloring as follows:

- if $x_t = true$ (resp. $false$), then $c(v_t) = 1$ (resp. $c(v_t) = 2$).
- all its copies $v_{t1}^r, v_{t2}^r, v_t^r$ for $r \in [\alpha_t]$ have color $c(v_t)$.

We now check that there are at least $10l + k$ fulfilled edges in G. Indeed, for each satisfied clause, the corresponding clause edge in $E(G)$ is fulfilled, then we have at least k such edges. For each variable x_t, and $r \in [\alpha_t]$, the v_t^r-graph has all 5 fulfilled edges. There are $2l$ such graphs which have totally $10l$ fulfilled edges. Therefore, c fulfills at least $k + 10l$ edges of G.

Conversely, let a coloring c which fulfills at least $10l + k$ edges of G.

Remark 78. We can easily check the following: all five edges in a v_t^r-graph are fulfilled if and only if its four vertices have the same colors.

Claim 78.1. *There is always a coloring c such that it fulfills all edges of v_t^r-graphs (for $t \in [N], r \in [\alpha_t]$).*

Proof of claim: Assume a coloring in which there is a v_t^r-graph which has a conflict edge. Observe that if v_t^r has at least two conflict edges or the clause edge at v_t^r is conflict, then we can fulfill all edges of v_t^r-graph by Remark 78 and obtain more fulfilled edges which contradicts to the maximality of c. Hence, v_r^t-graph contains only one conflict edge and the clause edge at v_t^r is fulfilled by c. We can fulfill all edges of this graph by changing the color of v_r^t, which leads to the following cases:

1. The clause edge $v_i^j u^j$ is still fulfilled, and we obtain more fulfilled edges which is a contradiction (Figure 6.2).

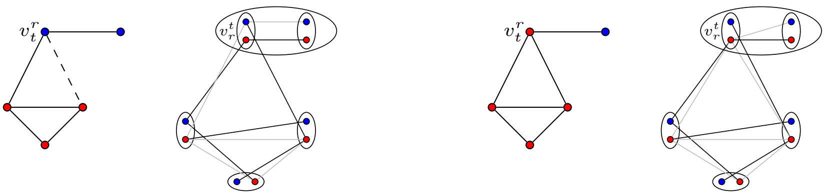


Figure 6.2: Corresponding to a color at each vertex, v_t^r-graph has one conflict (dashed) edge and the clause edge at v_t^r is fulfilled (left). After changing the color of v_t^r, the clause edge is still fulfilled and all edges of v_t^r-graph are now fulfilled (right).

2. The clause edge at v_t^r changes to a conflict edge, and we still have the same number of fulfilled edges of G (Figure 6.3).

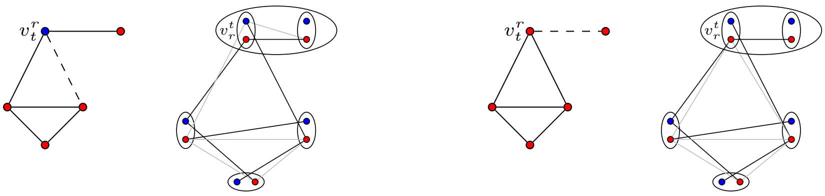


Figure 6.3: The case after changing the color of v_t^r, the clause edge becomes a conflict (dashed) edge and all edges of v_r^t-graph are now fulfilled (right).

Therefore, there is a coloring which fulfills all edges of v_t^r-graphs. ◁

From a coloring c satisfying Claim 78.1, we define ϕ by taking $\phi(x_t) = true$ (resp. $false$) if $c(v_t) = 1$ (resp. $c(v_t) = 2$). Since c fulfills all edges of v_t^r-graphs ($10l$ in total), then c fulfills at least k clause edges in G. Thus, ϕ satisfies at least k clauses of Φ.

2. $\mathcal{B}(2,3)$-FULFILL 2-COLORING is NP-hard.

We reduce from 3-SAT. Given a formula Φ of this problem with N variables x_t, $t \in [N]$ and l clauses C_i, $i \in [l]$, we construct a graph G and M_e for any $e \in E(G)$ as follows (see Figure 6.4 for an example).

- For each variable x_t, create a vertex $v_t \in V(G)$.
- For each clause C_i, we create a *clause gadget* X_i which contains six vertices $c_1^i, \ldots, c_6^i$. We add (ordered) edges $c_4^i c_5^i, c_5^i c_6^i, c_6^i c_4^i$ and for each edge uv of them, M_{uv} has three edges which are not $(u,1)(v,2)$. And add (ordered) edges $c_1^i c_4^i, c_2^i c_5^i, c_3^i c_6^i$ and for each edge uv of them, M_{uv} has three edges which are not $(u,2)(v,2)$. If $C_i = (\ell_1 \vee \ell_2 \vee \ell_3)$ where each ℓ_i is a literal of variable x_{t_i}, we then add a set

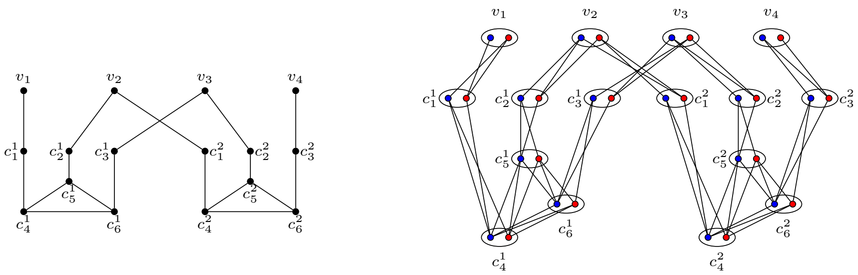


Figure 6.4: **An example of the construction in the reduction.** The graph G (left) which is associated to $\Phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$ and an example of graphs M_e for $e \in E(G)$. Note color 1 (resp. 2) is presented as blue (resp. red).

Y_i of *variable-clause edges* $v_{t_1}c_1^i, v_{t_2}c_2^i, v_{t_3}c_3^i$. For $j \in [3]$, $M_{v_{t_j}c_j^i}$ has three edges which are not $(v_{t_j}, 1)(c_j^i, 1)$ (resp. $(v_{t_j}, 2)(c_j^i, 1)$) if $\ell_j = x_{t_j}$ (resp. $\ell_j = \bar{x}_{t_j}$). For conveniences, for a clause C_i, the edge set which contains all edges of X_i and Y_i is called *clause-edge set*, denoted by S_i.

We can easily check by hand the following.

Claim 78.2. *For each clause C_i, a coloring fulfills no edges in Y_i if and only if it can fulfill at most three edges in S_i. Furthermore, there exists a coloring which fulfills four edges in S_i in which at least one of them is in Y_i.*

We shall show that there is ϕ satisfying Φ if and only if there is a 2-coloring of G with $4l$ fulfilled edges. The idea is that a variable $\phi(x_t) = true$ (resp. $false$) when $c(v_t) = 1$ (resp. $c(v_t) = 2$), and a literal $\ell_j = x_t$ (resp. $\ell_j = \bar{x}_t$) is true in a clause C_i when the edge $v_t c_j^i \in Y_i$ is fulfilled and then v_t is colored 1 (resp. 2).

Assume that there is an assignment ϕ satisfying Φ. We define a coloring c of G by taking $c(v_t) = 1$ if $\phi(x_t) = true$ or $c(v_t) = 2$ if $\phi(x_t) = false$, and we complete c such that if C_i is true, then c fulfills four edges of S_i which includes the variable-clause edge of every true literal in C_i; otherwise, c fulfills three edges in S_i and none of them is in Y_i. This can be done by Claim 78.2. Since there are l clauses of Φ which are true, then c fulfills $4l$ edges.

Conversely, assume that we have a 2-coloring c of G with $4l$ fulfilled edges. By Claim 78.2, c must fulfill at least one edge in Y_i for any clause C_i. As a consequence, we obtain ϕ by taking $\phi(x_t) = true$ (resp. $false$) if $c(v_t) = 1$ (resp. 2), hence ϕ satisfies Φ. $\square$

Remark 79. MAX CUT can be solved in polynomial time for planar graphs [Had75]. In contrast, $\mathcal{B}(2,3)$-FULFILL 2-COLORING is NP-complete on planar graphs by reducing from PLANAR 3-SAT which is NP-complete [Lic82] with the same reduction in Theorem 77-2.

When $k \geq 3$, we use reductions from k-COLORING to establish the following.

Theorem 80. *Let $k \geq 3$ be an integer. If $1 \leq m \leq k^2 - 3$ then $\mathcal{B}(k,m)$-*CONFLICT *k-*COLORING *is* NP*-complete, and if $m = \{k^2-1, k^2-2\}$ then it is polynomial-time solvable.*

The proof is separated to several cases depending of m.

Polynomial cases when $m \in \{k^2-1, k^2-2\}$. Given a graph G, we shall find a conflict k-coloring c of G if it exists.

If $m = k^2 - 1$, in order to fulfill an edge $uv \in E$, we have to take $c(u) = i, c(v) = j$ for the only edge $(u,i)(v,j) \notin M_{uv}$. If we can assign a color to every vertex, then we obtain a conflict k-coloring of G. Otherwise, G is not conflict colorable.

If $m = k^2 - 2$, the problem can be solved as follows. Initially, let $\tilde{c}(v) = \emptyset$ for every $v \in V$. We sequentially apply the following rules for all vertices.

Rule 1: For every edge $uv \in E$, if there is $i \in [k]$ such that the two edges $(u,i)(v,j_1), (u,i)(v,j_2) \notin M_{uv}$, then $\tilde{c}(u) := \tilde{c}(u) \cup \{i\}$.

If there is a vertex v such that $|\tilde{c}(v)| \geq 2$, then there is no coloring c because v needs two colors to fulfill all its incident edges. Otherwise, let $S = \{v \in V \mid |\tilde{c}(v)| = 1\} \subseteq V$.

We sequentially apply the following for all vertices $V \setminus S$.

Rule 2: For every vertex v with all its neighbors $\{u_1, \ldots, u_q\}$ and for every color $i \in [k]$ such that $(v,i)(u_j, c_j) \notin M_{vu_j}$ for all $j \in [q]$ and some $c_j \in [k]$, then we set $\tilde{c}(v) := \tilde{c}(v) \cup \{i\}$.

Observe that $|\tilde{c}(v)| \leq 2$ for every $v \in V$. If there is a vertex $v \in V$ such that $\tilde{c}(v) = \emptyset$, then there is no conflict coloring for G because there is no color for v that allows to fulfill all its incident edges. Without loss of generality, we assume that $G[V \setminus S]$ is connected (otherwise, we repeat the following for all the connected components). There are two cases:

(i) If there is a vertex $v \in V \setminus S$ such that $|\tilde{c}(v)| = 1$, let $(v_1 = v, v_2, \ldots, v_t)$ be the ordering obtained by a breadth-first search algorithm in $G[V \setminus S]$. By construction of S, and $V \setminus S$, this ordering gives a unique color for all vertices of $V \setminus S$, that is $c(v_1) = \tilde{c}(v_1)$, and $c(v_i) = c_i$ with c_i the unique color such that $(v_{i-1}, c(v_{i-1}))(v_i, c_i) \notin M_{v_{i-1}v_i}$ for every $i \in \{2, \ldots, t\}$. Finally, once the colors $c(v)$ have been determined for all vertices $v \in V$, we have to verify if all the edges of E are fulfilled.

(ii) If $|\tilde{c}(v)| = 2$ for every $v \in V \setminus S$, take any vertex $v \in V \setminus S$ and fix one color $c(v) \in \tilde{c}(v)$ and apply the procedure above in (i). If there is a coloring that fulfills all edges, then it terminates. Otherwise, apply the procedure in (i) for the other color in $\tilde{c}(v)$. Hence either we find a conflict k-coloring or G has no such a coloring.

The NP-completeness when $m \in [k^2-3]$. We denote by $[a,b]$ for $a \leq b$ the set of integers $\{a, a+1, \ldots, b\}$. For an edge uv, we reuse the notation K^k_{uv} which is the complete bipartite with the vertex set $V(K^k_{uv}) = \{(u,i), i \in [k]\} \cup \{(v,j), j \in [k]\}$ and

$E(K_{uv}^k) = \{(u,i)(v,j), i,j \in [k]\}$.
We reduce from k-COLORING and the proof includes the three following cases.

Case 1: $\frac{k}{2} \leq m \leq (k-1)^2$:
Given a graph $G = (V, E)$ of k-COLORING, we construct, for $\mathcal{B}(k,m)$-CONFLICT k-COLORING, a graph $G' = (V', E')$ and a k-SoC $\mathcal{M}$ as follows (Figure 6.5):

- For each $v \in V$, we add to V' a vertex v', and for each edge $uv \in E$, add k vertices $a_1^{uv}, \ldots, a_k^{uv}$. We set $E'_{uv} = \{ua_i^{uv}, va_i^{uv} \mid i \in [k]\}$ and add it to E'.
- For any $uv \in E$ and $i \in [k]$, $M_{ua_i^{uv}}$ contains $t = \min\{m, k-1\}$ edges $(u,i)(a_i^{uv}, j)$ for $j \in [t]$ and $t' = m - (k-1)$ (if $t' > 0$) edges of B_u where B_u is the complete bipartite graph whose vertex sets are $\bigcup_{j \in [k]\setminus\{i\}} \{(u,j)\}$ and $\bigcup_{j\in[2,k-1]} \{(a_i^{uv}, j)\}$. The conflict graph $M_{va_i^{uv}}$ contains t edges $(v,i)(a_i^{uv}, k), \ldots, (v,i)(a^{uv}, k-t+1)$ and t' (if $t' > 0$) edges of B_v where B_v is the complete bipartite graph whose vertex sets are $\bigcup_{j\in[k]\setminus\{i\}} \{(v,j)\}$ and $\bigcup_{j\in[2,k-1]} \{(a_i^{uv}, j)\}$.

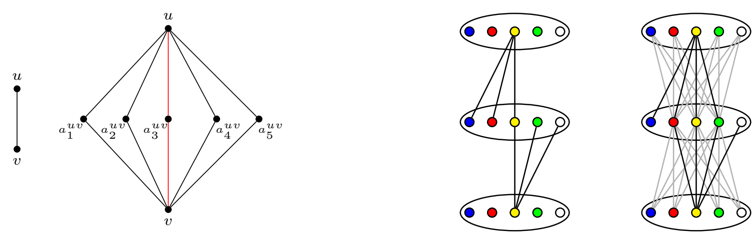


Figure 6.5: **An example of the construction when** $k = 5$**.** An edge $uv \in E$ and its set E'_{uv} (left). The conflict k-graphs of red edges ua_3^{uv}, va_3^{uv} when $m = 3$ (third graph); and if $m > 4$, there are $t = 4$ black edges and the $t' = m - k + 1$ other edges are chosen among grey edges (rightmost).

Observe that by the construction of the set $\mathcal{M}$, the value m is at most $(k-1)^2$, because for any edge $uv \in E$ and $i \in [k]$, $M_{ua_i^{uv}}$ and $M_{va_i^{uv}}$ contains at most $k - 1 + |E(B_u)| = k - 1 + |E(B_u)| = (k-1)^2$ edges.

We shall prove that G is k-colorable if and only if G' is $\mathcal{M}$-conflict colorable. Assume that c is a k-coloring of G, then we can construct a coloring c' of G' as follows: $c'(v') = c(v)$ for any $v \in V$, $c'(a_i^e) = k$ for all $e \in E, i \in [k]$. We can check easily that c' fulfills all edges of G'.

Conversely, there is an $\mathcal{M}$-conflict coloring c' of G'. Observe that if $c'(u) = i$ (for some $i \in [k]$), then $c'(a_i^{uv}) \in [t,k]$. It implies that $c'(v) \neq i$ because otherwise $t \geq k/2$, so va_i^{uv} is not fulfilled . Thus, $c'(u) \neq c'(v)$ and we then obtain a coloring c of G by taking $c(v) = c'(v')$ for every $v \in V$.

Case 2: $1 \leq m < \frac{k}{2}$:
We need the following lemma.

Lemma 81. *There is a graph family $\{G(k,m)\}$ for any $k, m \geq 1$ such that $G(k,m)$ is not $\mathcal{B}(k,m)$-conflict k-colorable.*

Proof. Given $m \geq 1$, we shall construct a family $\{G(k, m)\}$ and a k-SoC $\mathcal{M}$ by induction on k.

Let $G(k, m) = (V(k, m), E(k, m))$. Consider the following cases.

(i) For $m = k^2$, let $G(k, m)$ on 2 vertices a_1, a_2 and it has one edge a_1a_2. And let the conflict k-graph $M_{a_1a_2} = K^k_{a_1a_2}$ (Figure 6.6).

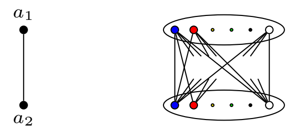


Figure 6.6: **Graph** $G(k, k^2)$**.**

(ii) For $k^2 > m \geq k$, $G(k, m)$ has $k + 1$ vertices $a, a_1, \dots, a_k$ and k edges aa_i for all $i \in [k]$. For each of them, the conflict k-graph M_{aa_i} contains k edges $(a, i)(a_i, j)$ for all $j \in [k]$, and $m - k$ other edges are chosen among the other edges of $K^k_{aa_i}$ (Figure 6.7).

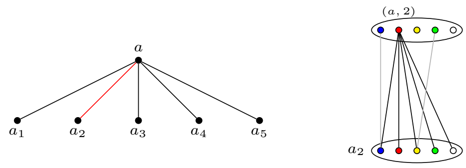


Figure 6.7: **An example of graph** $G(5, m)$ **with 5 colors.** The conflict 5-graph of red edge aa_2, when $m = 5$, M_{aa_2} contains five black edges $(a, 2)(a_2, i)$ for all $i \in [5]$; and when $m = 7$, then M_{aa_2} contains these black edges and two grey edges which are in the remaining edges of $K^5_{aa_2}$.

(iii) For $k > m$, $G(k, m)$ is build as follows (Figure 6.8). We add $k + 1$ vertices $a, a_1, \dots, a_k$ to $V(k, m)$, and add k edges aa_i for all $i \in [k]$ to $E(k, m)$. Now, for each vertex a_i, add a set X_i of $|G(k - m, m)| - 1$ new vertices, and let a copy C_i of the graph $G(k - m, m)$ on $a_i \cup X_i$. Then, connect a to all vertices of this copy C_i.

Now, we construct a set of conflict k-graphs for $G(k, m)$. For each edge ab (for every $b \in C_i, 1 \leq i \leq k$), the conflict k-graph M_{ab} contains m edges $(a, i)(b, j)$ for all $j \in [m]$. In each copy C_i, we apply $(k - m)$-SoC $\mathcal{M}'$ of graph $G(k - m, m)$ for $k - m$ colors $\{m + 1, \dots, k\}$, if $m \geq (k - m)^2$, then the conflict $(k - m)$-graph M_{uv} for $uv \in C_i$ contains the complete bipartite on two sets $\{(u, m + 1), \dots, (u, k)\}, \{(v, m + 1), \dots, (v, k)\}$ and $m - (k - m)^2$ remaining edges are taken in K^m_{uv}.

Trivially, graphs in case (i) have no $\mathcal{M}$-conflict coloring.

In case (ii), $G(k, m)$ has no conflict coloring, since choosing any color $i \in [k]$ for a implies that edge aa_i is not fulfilled.

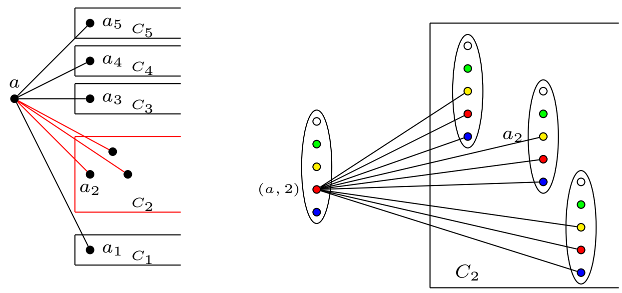


Figure 6.8: **An example of graph** $G(5,3)$**.** A copy $C_i = G(2,3)$ for all $i \in [5]$. The conflict k-graph of red edges and copy C_2 (right): vertex $(a,2)$ connects threes first vertices $(b,1),(b,2),(b,3)$ (presented as blue, red, yellow) and the set 2-SoC of $G(2,3)$ is applied to vertices $(b,4),(b,5)$ (presented as green, white) for any $b \in V(C_2)$.

In case (iii), by the construction, colors of $a_1,\ldots,a_k$ are not in $[m]$. However, if we choose a color for each a_i in $[m+1,k]$, then it can not fulfill all edges of the copy C_i because $G(k-m,m)$ is not conflict colorable with its $(k-m)$-SoC. Thus, the graphs of this case have no conflict coloring. $\square$

Now, we give a reduction from k-COLORING problem to $\mathcal{B}(k,m)$-CONFLICT k-COLORING for $1 \le m < k/2$.

Given a graph $G=(V,E)$, we construct a graph $G'=(V',E')$ and a k-SoC $\mathcal{M}$ as follows.

- Let A be a copy of V and add A to V'. For each edge $uv \in E$ with $u',v' \in A$ copies of u,v, we add k copies $C_1^{uv},\ldots,C_k^{uv}$ of the graph $G(k-2m,m)$, and add edges from each u',v' to all vertices in these k copies. Let $V'_{uv} = \{u',v',V(C_1^{uv}),\ldots,V(C_k^{uv})\}$.
- For any induced subgraph $G'[V'_{uv}]$, $u' \in A$, $a \in V(C_i^{uv})$ and $i \in [k]$, the conflict k-graph $M_{u'a}$ has m edges $(u,i)(a,1),\ldots,(u,i)(a,m)$; and the conflict k-graph $M_{v'b}$ for any $b \in V(C_j^{uv})$, $j \in [k]$ has m edges $(v,j)(b,k),\ldots,(v,j)(b,k-m+1)$. In each C_i^{uv}, we apply the $(k-2m)$-SoC of the graph $G(k-2m,m)$ for $k-2m$ colors $[m+1,k-m]$.

If G has a k-coloring c then we have c' for G' as follows: for each V'_{uv}, let $c'(v')=c(v)$, $c'(u') = c'(a) = c(u)$ for all $a \in \bigcup_{i\in[k]\setminus c(u)} V(C_i^{uv})$ and $c'(a) \in [k,k-m+1]$ for all $a \in V(C_{c(u)}^{uv})$. Hence, c' is a conflict coloring of G'.

Conversely, assume G' has a k-coloring c', then we prove that $c'(u') \ne c'(v')$ for all $u',v' \in A \cap V'_{uv}$. Observe that for any $a \in V(C_{c'(u')}^{uv})$, its color is not in $[k,k-m+1]$ because of the construction of $C_{c(u)}^{uv}$. Moreover, if $a \in C_i^{uv}$ has color $j \notin \{m+1,\ldots,k-m\}$, then (a,j) is either adjacent to (u',i) in the conflict k-graph M_{ua} or (v,i) in M_{va} for $i \in [k]$. Thus $c'(u') \ne c'(v')$. Therefore, we take $c(v) = c'(v')$ for all v' the copy of v and it is a k-coloring of G.

Case 3: $(k-1)^2 < m \leq k^2 - 3$.

We use an induction on $k^2 - m$ as follows. The base case is when $k = 3$ and $1 \leq m \leq 6$. From these two cases above, it suffices to prove for $m \in \{5, 6\}$.

Claim 81.1. *$\mathcal{B}(3, m)$-Conflict k-Coloring is NP-complete for $m = \{3, 4, 5, 6\}$.*

The inductive result is obtained by the following.

Claim 81.2. *If $\mathcal{B}(k, m)$-Conflict k-Coloring is NP-complete, then so is $\mathcal{B}(k', m')$-Conflict k'-Coloring for $k' > k$ and $k'^2 - m' = k^2 - m$.*

Observe that if $\mathcal{B}(k, m)$-Conflict k-Coloring is NP-complete for $1 \leq m \leq (k-1)^2$, then so is $\mathcal{B}((k+1), m')$-Conflict k-Coloring by Claim 81.2 for m' such that $m' = 2k + 1 + m$, thus $2k + 2 \leq m' \leq (k+1)^2 - 3$. Therefore, we have the following.

Theorem 82. *$\mathcal{B}(k, m)$-Conflict k-Coloring is NP-complete for $k \geq 3$ and $(k-1)^2 < m \leq k^2 - 3$.*

It remains to prove Claim 81.1 and Claim 81.2.

Proof of Claim 81.1. A reduction from 3-Coloring. Given a graph $G = (V, E)$ of this problem, we construct for $\mathcal{B}(3, m)$-Conflict k-Coloring, a graph $G' = (V', E')$ and a set of conflict 3-graphs $\mathcal{M}$ as follows (see Figure 6.9).

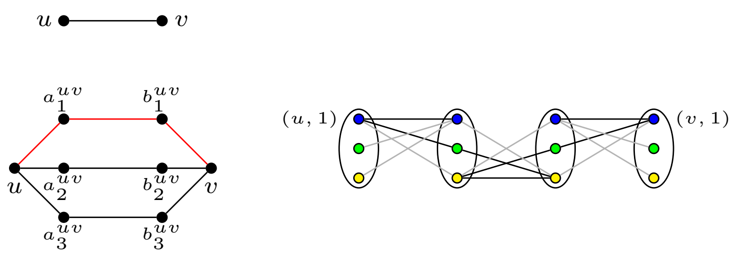


Figure 6.9: **Construction of the reduction.** For each edge $uv \in E(G)$, there is a gadget $\mathcal{G}^{uv}$. And an example on red edges is the structure of their conflict 3-graphs: it is obligatory that black edges are in conflict 3-graphs while gray ones are not. We obtain that $c(u) = 1$ if and only if $c(v) \neq 1$.

- For each edge $e = uv \in E$, add a gadget $\mathcal{G}^e$ of 8 vertices $u', v', a_1^e, b_1^e, a_2^e, b_2^e, a_3^e, b_3^e$ to V' where u', v' are respectively copies of u, v; and we add to E' edges $ua_i^e, a_i^e b_i^e, b_i^e v$ for all $i \in [3]$.
- For each induce subgraph $G'[\mathcal{G}^{uv}]$, we construct the conflict 3-graphs of edges $u'a_i^{uv}, a_i^{uv}b_i^{uv}, b_i^{uv}v'$ for any $i \in [3]$ as follows:
  - $M_{u'a_i^{uv}}$ contains two edges $(u', i)(a_i^{uv}, 1), (u', i)(a_i^{uv}, 2)$ and any $m-2$ other edges is in $K^3_{u'a_i^{uv}}$ except three edges $(u', i)(a_i^{uv}, 3), (u', j)(a_i^{uv}, 1)$ for $j \neq i$.
  - $M_{a_i^{uv}b_i^{uv}}$ contains three edges $(a_i^{uv}, 2)(b_i^{uv}, 3), (a_i^{uv}, 3)(b_i^{uv}, 2), (a_i^{uv}, 3)(b_i^{uv}, 3)$ and any $m-3$ other edges in $K^3_{a_i^{uv}b_i^{uv}}$ except two edges $(a_i^{uv}, 1)(b_i^{uv}, 3), (a_i^{uv}, 3)(b_i^{uv}, 1)$.

– $M_{b_i^{uv}v'}$ contains two edges $(b_i^{uv},1)(v',i),(b_i^{uv},2)(v',i)$ and any $m-2$ other ones in $K^3_{b_i^{uv}v'}$ except three edges $(b_i^{uv},3)(v',i),(b_i^{uv},1)(v',j)$ for $j \neq i$.

By the construction of $\mathcal{M}$, each conflict graph $M_e \in M$ needs at least three conflict edges and at least three edges not in M_e. Thus, $3 \leq m \leq 6$.

We prove now G is 3-colorable if and only if there is c' which fulfills all edges of G'.

If G has a k-coloring c, then G' has also a conflict k-coloring by, for all $\mathcal{G}^{uv}$, taking $c'(u')=c(u), c'(v')=c(v), c'(a_i^{uv})=3, c'(b_i^{uv})=1$.

Conversely, let a coloring c' which fulfills all edges of G' has a k coloring. Observe that $c'(u) \neq c'(v)$ for any $u,v \in \mathcal{G}^{uv}$ because by the construction of $G'[\mathcal{G}^{uv}]$, if $c'(u')=c'(v')=i$, then three edges $ua_i^{uv}, a_i^{uv}b_i^{uv}, b_i^{uv}v$ are not all fulfilled. Therefore, we define a coloring c of G by taking $c(v)=c'(v')$ for any $v' \in E'$ copy of $v \in E$. $\triangleleft$

Proof of Claim 81.2. Let a graph G and a k-SoC $\mathcal{M}$ for $\mathcal{B}(k,m)$-CONFLICT k-COLORING. We will construct for $\mathcal{B}(k',m')$-CONFLICT k'-COLORING, a graph $G'=(V',E')$ and $\mathcal{M}'$, as follows:

- Let G' be a copy of G (*i.e.* V',E' are respectively copies of V and E.
- For each edge $u'v'$ of G', $M'_{u'v'}$ contains the following: a copy of the conflict graph M_{uv} for colors in $[k]$, the complete bipartite graph between two sets $\{(u,i) \mid i \in [k+1,k']\}$ and $\{(v,j) \mid j \in [k']\}$, and the complete bipartite graph between two sets $\{(u,j) \mid j \in [k']\}$ and $\{(v,i) \mid i \in [k+1,k]\}$.

If c be a coloring which fulfills all edges of G, then taking $c'(v')=c(v)$ for all $v' \in V'$ copy of $c \in V$. Thus, c' also fulfills all edges of G'.

Conversely, there is a coloring c' which fulfills all edges of G'. Consider that for any $u'v' \in E'$, (v,i) is adjacent to (u,j) for all $i \in [k+1,k']$ and $j \in [k']$, then $c'(v') \in [k]$ for all $v' \in V'$. Let c be a coloring of G by taking $c(v)=c'(v')$ for $v \in V$, thus c fulfills all edges of G'. $\triangleleft$

Therefore, we complete the proof of Theorem 80. It implies that $\mathcal{B}(k,m)$-FULFILL k-COLORING is NP-complete for $k \geq 3$ and $1 \leq m \leq k^2-3$. Moreover, by using the same construction in Claim 81.2, we make a reduction from $\mathcal{B}(2,4-i)$-FULFILL k-COLORING to $\mathcal{B}(k,k^2-i)$-FULFILL k-COLORING for $i \in \{1,2\}$. Thus $\mathcal{B}(k,k^2-i)$-FULFILL k-COLORING is NP-complete when $i \in \{1,2\}$. We obtain the following.

Corollary 83. $\mathcal{B}(k,m)$-FULFILL k-COLORING *is* NP-*complete for* $k \geq 3$ *and* $m \in [k^2-1]$.

Chapter 7

Algorithms

In this chapter, we investigate algorithm techniques to solve FULFILL 2-COLORING and DOMINO. Namely, the first section present an approximation algorithm for FULFILL 2-COLORING. And in the next section, we give an FPT algorithm parameterized by *treewidth* for solving MAX DOMINO, the optimization version of DOMINO and a similar one for MAX THRESHOLD CONFLICT COLORING which outputs the maximum threshold value such that there exists a coloring which fulfills all edges of G.

7.1 Approximation algorithm for FULFILL 2-COLORING using Semidefinite programming (SDP)

Observe that MAX CUT is a particular case of FULFILL 2-COLORING in which the conflict 2-graph of any edge uv contains two edges $(u,1)(v,1)$ and $(u,2)(v,2)$. Goemans and Williamson showed a 0.87856-approximation algorithm for MAX CUT based on SDP [GW95]. In this section, we show how this method can be adapted to construct an approximation algorithm for FULFILL 2-COLORING. We will prove the following result.

Theorem 84. FULFILL 2-COLORING *is 0.79607-approximable.*

Proof. Let $V = \{v_1, \ldots, v_n\}$ be the set of vertices of a given a graph G. Let $y_i \in \{-1, 1\}$ be a variable corresponding to v_i. Let y_0 be an additional variable which determines if $y_i = 1$ corresponds to color $c(v_i) = 1$ or 2. Precisely, $y_i = y_0$ if and only if v_i has color 1.

Let e be an edge of G. We shall define a formula $f(e)$ depending on M_e which is equal to 1 (resp. 0) if the edge is fulfilled (resp. conflict). M_e is one of the conflict 2-

graphs depicted in Figure 7.1: there are i and j such that $e = v_i v_j$ and v_i is the bottom vertex and v_j the top vertex.

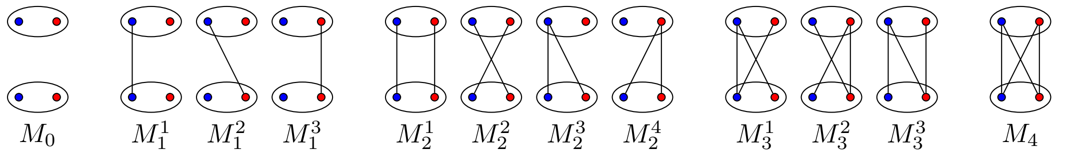


Figure 7.1: **The possible conflict 2-graphs (up to top-bottom symmetry).** Blue is color 1 and red is color 2.

Observe that if $M_e = M_0$, then e is fulfilled, so we set $f(e) = 1$, and if $M_e = M_4$, then e is never fulfilled, so we set $f(e) = 0$.

$f(e) = \dfrac{1 - y_0 y_i}{4} + \dfrac{1 - y_0 y_j}{4} + \dfrac{1 - y_i y_j}{4}$ if $M_e = M_1^1$;

$f(e) = \dfrac{1 - y_0 y_i}{4} + \dfrac{1 + y_0 y_j}{4} + \dfrac{1 + y_i y_j}{4}$ if $M_e = M_1^2$;

$f(e) = \dfrac{1 + y_0 y_i}{4} + \dfrac{1 + y_0 y_j}{4} + \dfrac{1 - y_i y_j}{4}$ if $M_e = M_1^3$;

$f(e) = \dfrac{1 - y_i y_j}{2}$ if $M_e = M_2^1$; $\quad f(e) = \dfrac{1 + y_i y_j}{2}$ if $M_e = M_2^2$;

$f(e) = \dfrac{1 - y_0 y_i}{2}$ if $M_e = M_2^3$; $\quad f(e) = \dfrac{1 + y_0 y_i}{2}$ if $M_e = M_2^4$;

$f(e) = \dfrac{1 - y_0 y_i - y_0 y_j + y_i y_j}{4}$ if $M_e = M_3^1$;

$f(e) = \dfrac{1 + y_0 y_i + y_0 y_j + y_i y_j}{4}$ if $M_e = M_3^2$;

$f(e) = \dfrac{1 + y_0 y_i - y_0 y_j - y_i y_j}{4}$ if $M_e = M_3^3$.

Note that if $M_e \in \{M_1^1, M_1^2, M_1^3\}$, then the formulas are the ones for MAX 2-SAT (where $c(u) = 1$ corresponds to u being *true* and $c(u) = 2$ to u being *false*), if $M_e \in \{M_2^1, M_2^2\}$, then the formulas are the ones for MAX CUT, and if $M_e \in \{M_3^1, M_3^2, M_3^3\}$, then the formulas are similar to those of MAX DICUT [GW95].

We formulate our problem as an integer programming problem (P) which maximizes $S = \sum_{e \in E(G)} f(e)$ subject to $y_i \in \{-1, 1\}$. Since in any case $f(e)$ is a nonnegative linear combination of $1 - y_i y_j$, $1 + y_i y_j$ and $1 - y_i y_j - y_i y_k + y_j y_k$, $1 + y_i y_j + y_i y_k + y_j y_k$, there are nonnegative coefficient a_{ij}, b_{ij}, c_{ijk} and d_{ijk} such that

$$S = \sum_{0 \le i,j,k \le n} a_{ij}(1 - y_i y_j) + b_{ij}(1 + y_i y_j) + c_{ijk}(1 - y_i y_j - y_i y_k + y_j y_k) + d_{ijk}(1 + y_i y_j + y_i y_k + y_j y_k).$$

We can relax (P) by replacing every y_i to a unit vector $x_i \in S_n$, the n-dimentional unit sphere. This results in the problem (P′) which maximizes

$$S' = \sum_{0 \le i,j,k \le n} a_{ij}(1 - x_i x_j) + b_{ij}(1 + x_i x_j) + c_{ijk}(1 - x_i x_j - x_i x_k + x_j x_k) + d_{ijk}(1 + x_i x_j + x_i x_k + x_j x_k)$$

subject to $x_i \in S_n$. This is a SDP problem with positive semidefinite matrix X with $X_{ij} = x_i x_j$. Using the method of Goemans and Williamson, we have the following algorithm.

1. Solve (P′)
2. Take a uniformly random vector r on the unit sphere S_n.
3. For $0 \leq i \leq n$, set $y_i = -1$ if $x_i r < 0$, and $y_i = +1$ otherwise.
4. For $1 \leq i \leq n$, set $c(v_i) = 1$ if $y_i y_0 = 1$ and $c(v_i) = 2$ otherwise.

y_i may be seen as the sign of $x_i r$. By Step 3 and 4, two vertices v_i and v_j have different colors if and only if $y_i \neq y_j$. This happens with probability $\frac{\theta_{ij}}{\pi}$ where θ_{ij} is the angle between x_i, x_j. Note that $x_i x_j = \cos\theta_{ij}$.

Let $S_a = \sum_{i,j} a_{ij}(1-y_iy_j)$, $S_b = \sum_{i,j} b_{ij}(1+y_iy_j)$, $S_c = \sum_{i,j,k} c_{ijk}(1-y_iy_j-y_iy_k+y_jy_k)$ and $S_d = \sum_{i,j,k} d_{ijk}(1+y_iy_j+y_iy_k+y_jy_k)$. We have $S = S_a+S_b+S_c+S_d$, so by linearity of the expectation $\mathbb{E}(S) = \mathbb{E}(S_a) + \mathbb{E}(S_b) + \mathbb{E}(S_c) + \mathbb{E}(S_d)$.

$$\begin{aligned}
\mathbb{E}(S_a) &= 2\sum_{i,j} a_{ij}\mathbb{P}(y_i \neq y_j) = 2\sum_{i,j} a_{ij}\frac{\theta_{ij}}{\pi} \\
&\geq \alpha \sum_{i,j} a_{ij}(1-\cos\theta_{ij}) = \alpha\sum_{i,j} a_{ij}(1-x_ix_j)
\end{aligned}$$

$$\text{where } \alpha = \min_{0\leq\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} > 0.87856$$

.

Similarly, $\mathbb{E}(S_b) \geq \alpha \sum_{i,j} b_{ij}(1+x_ix_j)$.

$$\begin{aligned}
\mathbb{E}(S_c) &= 4\sum_{i,j,k} c_{ijk}\mathbb{P}(y_i \neq y_j = y_k) \\
&\geq 4\sum_{i,j,k} c_{ijk}\frac{\theta_{ik}+\theta_{jk}-\theta_{jk}}{2\pi} \\
&\geq \beta\sum_{i,j,k} c_{ijk}(1-x_ix_j-x_ix_k+x_jx_k)
\end{aligned}$$

where $\beta = \min_{0\leq\theta\leq\arccos(-1/3)} \frac{2}{\pi}\frac{2\pi-3\theta}{1+3\cos\theta} > 0.79607$ as shown in [GW95].

Similarly, $\mathbb{E}(S_d) \geq \beta\sum_{i,j,k} d_{ijk}(1+x_ix_j+x_ix_k+x_jx_k)$.

Hence

$$\begin{aligned}
\mathbb{E}(S) &\geq \alpha\sum_{i,j}\Big(a_{ij}(1-x_ix_j)+b_{ij}(1+x_ix_j)\Big) \\
&\quad +\beta\sum_{i,j,k}\Big(c_{ijk}(1-x_ix_j-x_ix_k+x_jx_k)+d_{ijk}(1+x_ix_j+x_ix_k+x_jx_k)\Big) \\
&\geq \beta S'
\end{aligned}$$

But the x_i have been computed to give the maxi the optimization version ofmum value $\mathsf{OPT}(\mathsf{P}')$ of S', so $\mathbb{E}(S) \geq \beta\mathsf{OPT}(\mathsf{P}') \geq \beta\mathsf{OPT}(\mathsf{P})$. Therefore, the algorithm is a β-approximation of FULFILL 2-COLORING. □

7.2 Polynomial-time algorithms for graphs with bounded treewidth

In this section, we give a polynomial-time algorithm that solves the maximization version of DOMINO, called MAX DOMINO, when the treewidth of the graph is bounded. In a similar way, we can prove that for each fixed positive integer τ, there is a polynomial-time algorithm that solves MAX THRESHOLD CONFLICT COLORING, when the treewidth of the graph is at most τ. Given a graph G, integers k, q, and a set of weight functions w^k, MAX THRESHOLD CONFLICT COLORING consists in determining the maximum threshold λ for which there is a k-coloring that $\mathcal{M}^\lambda$-fulfills all edges of G.

Nice tree decomposition

A *tree decomposition* of a graph $G = (V, E)$ is a pair $\mathcal{T} = (T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_x \mid x \in V(T)\}$ is a family of subsets of V, called *bags*.

1. Every graph vertex $v \in V$ is in at least one bag;
2. For every edge $uv \in E$, there is at least one bag that contains both u and v;
3. For every graph vertex $v \in V$, the subgraph induced by the nodes whose bag contains v is a subtree T_v of T.

The *width* of a tree decomposition $\mathcal{T}$ is $\tau(\mathcal{T}) = \max_{B\in\mathcal{X}} |B| - 1$, and the *treewidth* of a graph G, denoted by $\text{tw}(G)$, is the minimum width among all the possible tree decompositions of G. Figure 7.2 depicts an example of a graph and a tree decomposition of width 2.

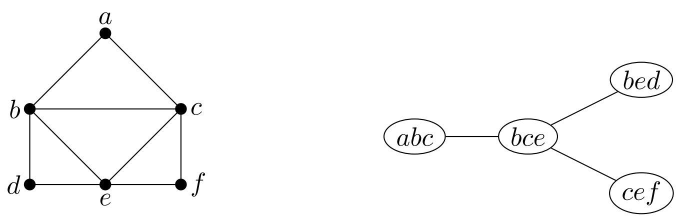


Figure 7.2: **A graph and a tree decomposition of this graph.**

In order to keep our algorithm and its analysis simple, we use a particular kind of tree decompositions called *nice tree decompositions*. A *nice tree decomposition* $\mathcal{T} = (T, \mathcal{X})$ is a tree decomposition such that T is rooted at a node r and satisfies the following conditions:

1. If $x \in V(T)$ is a leaf or the root, then $X_x = \emptyset$.

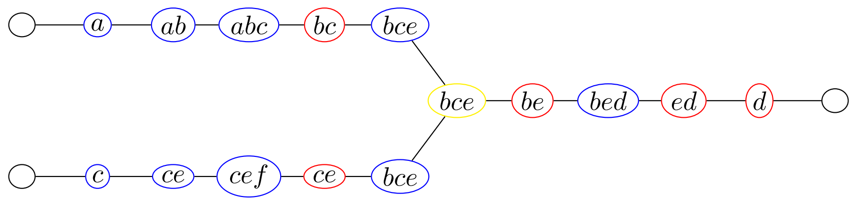


Figure 7.3: **The nice tree decomposition of the tree graph depicted in Figure 7.2.** The two leaves (leftmost nodes) and the root (rightmost node) are empty nodes, introduce nodes are blue, forget nodes are red and a join node is yellow.

2. Every other node x of T is one of the three following types:
   a) x is an *introduce node* if x has a unique child y in T and $X_x = X_y \cup \{v\}$ for some $v \in V(G) \setminus X_y$. We say that v is *introduced at* x.
   b) x is a *join node* if x has two children x_1 and x_2 in T and $X_x = X_{x_1} = X_{x_2}$.
   c) x is a *forget node* if x has a unique child y in T and $X_x = X_y \setminus \{v\}$ for some $v \in X_y$. We say that v is *forgotten at* x.

Figure 7.3 describes an example of nice tree decomposition of the graph depicted in Figure 7.2.
Note that any graph vertex is forgotten only once (by Condition 3 of a tree decomposition) but may be introduced several times.

Lemma 85 ([CVFK+15]). *Given a tree decomposition $\mathcal{T} = (T, \mathcal{X})$ of width at most τ of a graph G, one can compute in $O(\tau^2 \max(|V(G)|, |V(T)|))$ time a nice tree decomposition of width at most τ with at most $O(\tau.|V(G)|)$ nodes.*

7.2.1 Max Domino

Theorem 86. *Given a graph $G = (V, E)$ with treewidth at most τ, an integer $k \geq 1$, a set of weight functions w^k, the problem* Max Domino *can be solved in time $O(\tau^2 \cdot k^{\tau+2} \cdot |V|)$.*

In order to prove this theorem, we need some definitions and observations. Let $\mathcal{T} = (T, \mathcal{X})$ be a nice tree decomposition of G and let r be the root of T. Let x be a node of T. We denote by T_x the subtree of T rooted at x and containing all the descendants of x. We construct the graph $G_x = (V_x, E_x)$ as follows. $V_x = \bigcup_{y \in T_x} X_y$ and E_x is constructed as follows.

1. If x is a leaf of T, then $E_x = \emptyset$.
2. If x is an introduce node with child y, then $E_x = E_y$.
3. If x is a forget node with child y, then $E_x = E_y \cup \{vw \in E(G) \mid w \in X_x\}$ with $\{v\} = X_y \setminus X_x$.
4. If x is a join node of T with children x_1 and x_2, then $E_x = E_{x_1} \cup E_{x_2}$. Note that $V_x = V_{x_1} \cup V_{x_2}$.

By the construction of G_x, every edge of G_x has an endvertex that has been forgotten.

Given a node $x \in V(T)$ and a k-coloring $c_x : X_x \to [k]$, we define $S(x, c_x)$ as the value of an optimal solution for MAX DOMINO problem for the graph G_x with the additional constraint that $c(b) = c_x(b)$ for every $b \in X_x$. Formally,

$$S(x, c_x) = \max_{\substack{c:V_x \to [k] \\ \forall b \in X_x, c(b)=c_x(b)}} \sum_{uv \in E_x} w^k((u, c(u))(v, c(v))). \tag{7.1}$$

By convention we set $S(x, c_x) = 0$ when x is a leaf. We are now able to precisely design our dynamic programming algorithm, called MAX DOMINO ALGORITHM, that consists of computing $S(x, c_x)$ for every $x \in V(T)$ and every k-coloring c_x of X_x. Lemma 87 shows the formulas allowing such a computation.

Lemma 87. *Let x be a node of T and c_x a k-coloring of X_x.*

(a) If x is a join node with children x_1 and x_2, then $S(x, c_x) = S(x_1, c_x) + S(x_2, c_x)$.

(b) If x is an introduce node with child y, then $S(x, c_x) = S(y, c_y)$, where c_y is the k-coloring of X_y such that $c_y(v) = c_x(v)$ for all $v \in X_y$.

(c) If x is a forget node with child y, then

$$S(x, c_x) = \max_{c_y \in \mathcal{C}(c_x)} \left\{ S(y, c_y) + \sum_{\substack{w \in X_x \\ vw \in E_x}} w^k((v, c_y(v))(w, c_y(w)) \right\}$$

where $\{v\} = X_y \setminus X_x$ and $\mathcal{C}(c_x)$ is the set of k-colorings of X_y such that $c_y(b) = c_x(b)$ for all $b \in X_x$.

Proof. (a) Suppose x is a join node with children x_1 and x_2. We have $X_x = X_{x_1} = X_{x_2}$ and, by definition of a tree decomposition, any node in $V_{x_1} \setminus X_x$ (resp. $V_{x_2} \setminus X_x$) is not in X_{x_2} (resp. X_{x_1}). Thus $S(x, c_x) = S(x_1, c_x) + S(x_2, c_x)$.

(b) Suppose x is an introduce node with child y. Then by definition, $E_x = E_y$. So clearly $S(x, c_x) = S(y, c_y)$.

(c) Suppose x is a forget node with child y. Let $\{v\} = X_y \setminus X_x$. By definition $G_x = G_y \cup \{vw \in E(G) \mid w \in X_x\}$. Let c^* be a k-coloring of V_x such that $S(x, c_x) = \sum_{ab \in E_x} w^k((a, c^*(a))(b, c^*(b)))$ and let c_y be the restriction of c^* to X_y. Note that $c_y \in \mathcal{C}(c_x)$. Then

$$\begin{aligned} S(x, c_x) &= \sum_{ab \in E_y} w^k((a, c^*(a))(b, c^*(b))) + \sum_{\substack{vw \in E(G) \\ w \in X_x}} w^k((v, c^*(v))(w, c^*(w))) \\ &= \sum_{ab \in E_y} w^k((a, c^*(a))(b, c^*(b))) + \sum_{\substack{vw \in E(G) \\ w \in X_x}} w^k((v, c_y(v))(w, c_y(w))) \end{aligned}$$

Since c^* maximizes this sum and the second term is the same for all coloring whose restriction to X_y is c_y, we get $S(x, c_x) = S(y, c_y) + \sum\limits_{\substack{vw \in E(G) \\ w \in X_x}} w^k((v, c_y(v))(w, c_y(w)))$.

Therefore $S(x, c_x) = \max\limits_{c_y \in \mathcal{C}(c_x)} \left\{ S(y, c_y) + \sum\limits_{\substack{w \in X_x \\ vw \in E_x}} w^k((v, c_y(v))(w, c_y(w)) \right\}$ $\square$

Proof of Theorem 86. The algorithm is the following:

1. Compute a nice tree decomposition $\mathcal{T} = (T, \mathcal{X})$ of G with width at most τ.
2. Compute the value of $S(x, c_x)$ for all node x of T and all k-coloring c_x of X_x.
3. Return the maximum value of $S(r, c_r)$ over all k-colorings c_r of the bag X_r of the root r.

At Step 2, each value $S(x, c_x)$ is computed using the formulas of Lemma 87. If x is a leaf, a join node or an introduce node, then the value of $S(x, c_x)$ is obtained in constant time by the formula. If x is a forget node, then the algorithm takes the maximum over k values because $|\mathcal{C}(c_x)| = k$ (there are k possible colors for the vertex v of $X_y \setminus X_x$). The number of edge weights in the sum is upper-bounded by the size of X_x. Hence the value can be computed in $O(\tau \cdot k)$ time.
Now for each node x, there are at most $k^{\tau+1}$ possible k-colorings of X_x, so there are at most $k^{\tau+1} \cdot |V(T)|$ values $S(x, c_x)$, so at most $O(\tau \cdot k^{\tau+1} \cdot |V|)$ values by Lemma 85. As each of these values is computed in $O(\tau \cdot k)$ time, Step 2 runs in $O(\tau^2 \cdot k^{\tau+2} \cdot |V|)$ time.

Clearly Step 1 and 3 can be done faster than Step 2, so the total running time of our algorithm is $O(\tau^2 \cdot k^{\tau+2} \cdot |V|)$. $\square$

7.2.2 Max Threshold Conflict Coloring

In this section, we will prove the following theorem.

Theorem 88. *Given a graph $G = (V, E)$ with treewidth at most τ, an integer $k \geq 1$, a set of weight functions w^k, the problem* Max Threshold Conflict Coloring *can be solved in time $O(\tau^2 \cdot k^{\tau+2} \cdot |V|)$.*

Given a node $x \in V(T)$, a k-coloring $c_x : X_x \to [k]$, we define $\lambda(x, c_x)$ as the maximum threshold for Max Threshold Conflict Coloring for graph G_x with the constraint that $c(b) = c_x(b)$ for all $b \in X_x$. Observe that, the objective of the problem is to maximize $\min\limits_{uv \in E_x} w^k((u, c(u))(v, c(v)))$. Formally, we have

$$\lambda(x, c_x) = \max_{\substack{c : V_x \to [k] \\ \forall b \in X_x, c(b) = c_x(b)}} \min_{uv \in E_x} w^k((u, c(u))(v, c(v))). \tag{7.2}$$

We set $\lambda(x, c_x) = +\infty$ if x is a leaf. We are now able to precisely design our dynamic programming algorithm, called Max Threshold Conflict Coloring

ALGORITHM, that consists of computing $\lambda(x, c_x)$ for every $x \in V(T)$ and every k-coloring c_x of X_x. Lemma 89 shows the formulas allowing such a computation.

Lemma 89. *Let x be a node of T and c_x a k-coloring of X_x.*

(a) *If x is a join node with children x_1 and x_2, then $\lambda(x, c_x) = \min_{i\in\{1,2\}} \{\lambda(x_i, c_x)\}$.*

(b) *If x is an introduce node with child y, then $\lambda(x, c_x) = \lambda(y, c_y)$, where c_y is the k-coloring of X_y such that $c_y(v) = c_x(v)$ for all $v \in X_y$.*

(c) *If x is a forget node with child y, then*

$$\lambda(x, c_x) = \max_{c_y\in\mathcal{C}(c_x)} \min \left\{ \lambda(y, c_y), \min_{\substack{w\in X_x \\ vw\in E_x}} w^k((v, c_y(v))(w, c_y(w))) \right\}$$

where $\{v\} = X_y \setminus X_x$ and $\mathcal{C}(c_x)$ is the set of k-colorings of X_y such that $c_y(b) = c_x(b)$ for all $b \in X_x$.

Proof. Similarly as in Lemma 87, (a) and (b) are easily obtained.

(c) For x a forget node with child y, let $v = X_y \setminus X_x$.

$$\begin{aligned}
\lambda(x, c_x) &= \max_{\substack{c:V_x\to[k] \\ \forall b\in X_x, c(b)=c_x(b)}} \min_{uv\in E_x} w^k((u, c(u))(v, c(v))) \\
&= \max_{\substack{c:V_x\to[k] \\ \forall b\in X_x, c(b)=c_x(b)}} \min \left\{ \min_{uv\in E_y} w^k((u, c(u))(v, c(v))), \min_{\substack{w\in X_x \\ vw\in E_x}} w^k((v, c_y(v))(w, c_y(w))) \right\} \\
&= \max_{c_y\in\mathcal{C}(c_x)} \min \left\{ \lambda(y, c_y), \min_{\substack{w\in X_x \\ vw\in E_x}} w^k((v, c_y(v))(w, c_y(w))) \right\}.
\end{aligned}$$

□

We are now able to prove Theorem 88 (the proof is similar to that of Theorem 86).

Proof of Theorem 88. The algorithm is the following:

1. Compute a nice tree decomposition $\mathcal{T} = (T, \mathcal{X})$ of G with width at most τ.
2. Compute the value of $\lambda(x, c_x)$ for all node x of T and all k-coloring c_x of X_x.
3. Return the maximum value of $\lambda(r, c_r)$ over all k-colorings c_r of the bag X_r of the root r.

The running time is obtained by the same explanation of which in Theorem 86.

□

Chapter 8

Experiments

8.1 Geometric random instances of conflict coloring

As a test set, we consider instances associated with Voronoi diagrams in the plane. This geometric setting is especially interesting for several reasons: (i) the number of nodes (cells in the Voronoi diagram) can be tuned, and so is the number of colors (ii) the combinatorial structure of the Voronoi diagram can be used to infer local thresholds/constraints.

Given a set of 2D data points, recall that the Voronoi diagram (VD) is the partition of the plane into convex regions, such that all points in one region have a data point as nearest neighbor [BY98]. The 2D VD has a dual structure known as the Delaunay triangulation (DT), which (generically) consists of vertices, edges and triangles.

Consider the DT of a dataset P of N points drawn uniformly at random in a square of side $\sqrt{N} * 100$. This ensures that the average density of points is the same for any value of N, see Remark 91 below. Using this geometric setting, we define geometric instances of the conflict coloring problem as follows (Figure 8.1, Figure 8.2):

- (Nodes) The nodes of the graph correspond to the data set P of N points. More precisely, we retain as vertices those data points have a *finite Voronoi region* (there are N' such vertices), or equivalently, which do not lie on the convex hull of the data points.
- (Conformations/colors) Consider a data point $p \in P$ and its Voronoi region $V_r(p)$ – the index r stands for *reference*. Let v be a *Voronoi vertex* of $V_r(p)$. (Note that all such points are finite since p is not on the convex hull.) We take a random perturbation of v in $\mathcal{D}(v,\delta) \setminus V_r(p)$ where $\mathcal{D}(v,\delta)$ is the disc centered at v and radius δ. The set of all such perturbed Voronoi vertices defines a random

(perturbed) polygon for p. The set of all perturbed regions (polygons) for point p, denoted $C[p]$, defines its colors.

- (Conflict) Consider two Voronoi regions $V(p) \in C[p]$ and $V(q) \in C[q]$ of two neighboring points in DT. We compute the surface area of the intersection of these two regions, denoted $ia(V(p), V(q))$ for *intersection area*. The regions $V(p)$ and $V(q)$ are termed in conflict if $ia(V(p), V(q)) \geq \tau_c$, for some threshold τ_c.

Summarizing, a random instance is defined by the four tuple: $I = (P, \delta, C, \tau_c)$ with $P = \{p_1, \ldots, p_N\}$ a set of N points, δ the radius used to perturb the Voronoi vertices, c the number of colors/randomly perturbed Voronoi regions, and τ_c the threshold to define conflicts.

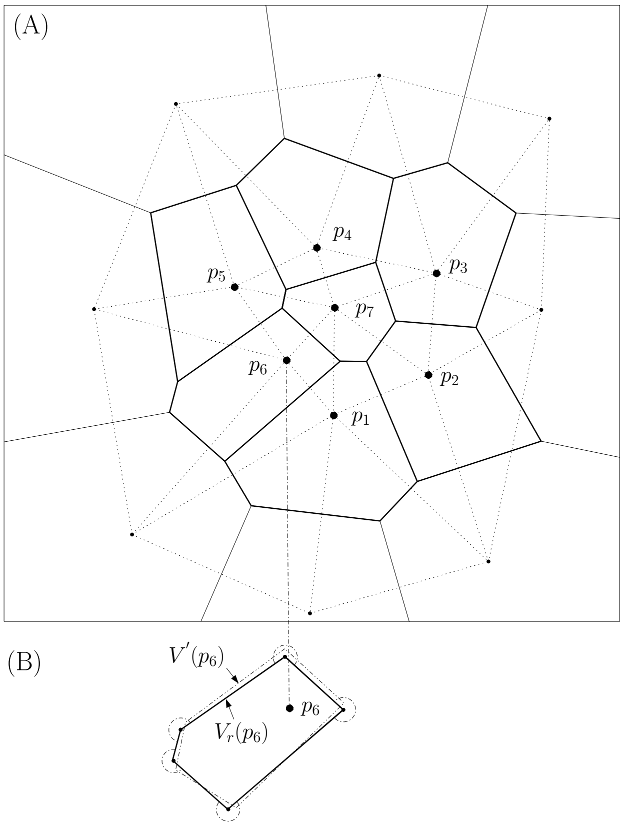


Figure 8.1: **Random instance of color matching defined from perturbations of a 2D Voronoi diagram. (A)** In a random 2D Voronoi diagram, points with a finite Voronoi cell (bold cells) a retained – $p_1, \ldots, p_7$ in the example; each such point corresponds to the vertices in the considered graph. The dotted line segments represent the Delaunay triangulation. **(B)** To create the *colors* associated with a node p of the graph, for each Voronoi vertex v of the Voronoi cell $V_r(p)$, a perturbed point of p is uniformly chosen at random in $\mathcal{D}(v, \delta) \setminus V_r(p)$.

Remark 90. (Implementation notes.) Recall that a polygon is termed *simple* if two of its edges do not intersect, except possibly at their endpoints. In this work, we use polygon representation provided by the Computational Geometry Algorithm Library

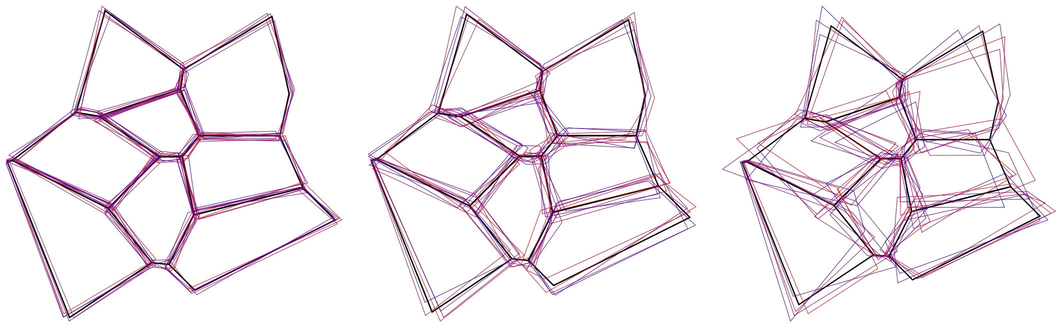


Figure 8.2: **Instances corresponding to the construction of Fig. 8.1 upon varying the perturbation radius δ.** Perturbed random polygons of the finite Voronoi regions (thick black polygons) given a data points of size $N = 15$ after generating with perturbed radius $\delta = 0.5$ (left), $\delta = 1$ (middle) and $\delta = 2$ (right) respectively and $c = 5$.

[cga], see CGAL-doc. For the sake of simplicity, in sampling conformations, we retain simple polygons only.

8.2 Parameters monitored

A random instance and a threshold τ_c yield a set of conflicts. We wish to get insights on the relationship between the value of τ_c derived from *local sufficient conditions*, and the existence of global solutions.

Local constraints based on min intersection areas. We use the topological information contained in the DT to derive sufficient conditions on the existence of *local* solutions. More specifically, we consider three nested local levels:

- **Edges.** For an edge p_ip_j of the DT, the reference regions $V_r(p_i)$ and $V_r(p_j)$ share a Voronoi edge, and the perturbed regions may overlap. For this edge, we compute the minimum intersection area $\tau_2(p_ip_j)$ of their perturbed regions, which ensures the existence of a local solution, this can be formalized as follows:

$$\tau_2(p_ip_j) = \min_{\substack{\boldsymbol{V_i \in C[p_i]} \\ \boldsymbol{V_j \in C[p_j]}}} ia(V_i, V_j). \tag{8.1}$$

  The max value obtained over all edges is denoted $\tau_2^{\text{Max-Min}}$.

- **Triangles.** Consider a triangle $(p_ip_jp_k)$ (face) in DT. In a manner analogous to edges, we compute the minimum value $\tau_3(p_ip_jp_k)$ required have a local solution for this triple. Formally:

$$\tau_3(p_ip_jp_k) = \min_{\substack{\boldsymbol{V_i \in C[p_i]} \\ \boldsymbol{V_j \in C[p_j]} \\ \boldsymbol{V_k \in C[p_k]}}} \max\{ia(V_i, V_j), ia(V_i, V_k), ia(V_j, V_k)\}. \tag{8.2}$$

  The max value obtained over all triangles is denoted $\tau_3^{\text{Max-Min}}$.

- **Stars.** Finally, we consider the star centered at each data point p_i, that contains p_i and its set of neighbors $N(p_i)$ in the DT with finite dual Voronoi cell. We compute the minimum value $\tau_*(p_i)$ of the threshold, which ensures the existence of a local solution of the star. Formally:

$$\tau_*(p_i) = \boldsymbol{\min_{V_i \in C[p_i]}} \max_{p_j \in N(p_i)} \min_{V_j \in C[p_j]} ia(V_i, V_j). \tag{8.3}$$

  The max value obtained over all stars is denoted $\tau_*^{\text{Max-Min}}$.

The values $\tau_2(p_ip_j)$, $\tau_3(p_ip_jp_k)$ and $\tau_*(p_i)$ provide local necessary conditions. Observe that, in order to have a global solution for an instance, the threshold δ_c must be at least all these local values. We therefore define

$$\tau_{\text{loc.}}^{\text{Max-min}} = \max\{\tau_2^{\text{Max-Min}}, \tau_3^{\text{Max-Min}}, \tau_*^{\text{Max-Min}}\}. \tag{8.4}$$

Local constraints based on average intersection areas. In the previous equations, one uses the most stringent constraint for an edge / triangle / star. Consider the case where one increases the number of colors: the min values observed for an edge / triangle / star is expected to decrease. Which means in turn that in increasing the number of colors, fewer solutions are expected overall. We also consider a variant of the problem where the local thresholds are defined from mean intersection areas. Replace the (bold) min by mean in Equations 8.1, 8.2, 8.3 yield the following:

$$\tau_{\text{loc.}}^{\text{Max-Mean}} = \max\{\tau_2^{\text{Max-Mean}}, \tau_3^{\text{Max-Mean}}, \tau_*^{\text{Max-Mean}}\}. \tag{8.5}$$

Let m be some multiplicative factor – that will differ for τ_c^{min} and τ_c^{mean}. Using the previous quantities, we define global values of τ_c as follows:

$$\begin{cases} \tau_c^{\text{min}} = m \times \tau_{\text{loc.}}^{\text{Max-Min}}, \\ \tau_c^{\text{mean}} = m \times \tau_{\text{loc.}}^{\text{Max-Mean}}. \end{cases} \tag{8.6}$$

Remark 91. In sampling data points in a square of side $\sqrt{N} * 100$, the expected area for each VD region is 10^4, the threshold values are in unit and we may mention also in brackets the fraction of a threshold value over the expected area, for example $\tau_{\text{loc.}} = 100$ (1% of the expected area of a VD region).

8.3 Results

Setup. We consider two complementary analysis: *the fraction of solutions in the entire search space, and the existence of at least one solution.*

The following values are used in our experiments. Random instances are generated with parameters $N = 15, \delta = 1$.

For each instance, we compute the values of τ_c^{min} and τ_c^{mean}. We retain random instances whose $\tau_{\text{loc.}}^{\text{Max-Min}} \leq 100$ (thus at most 1% of the expected area of a VD polygon). The number of colors used in the experiments varies in the set $C \in \{5, 10, 15, 50, 100\}$. The multiplicative factors vary $m \in m_1 = \{1.01, 1.5, 2, 3, \ldots, 10\}$ for τ_c^{min} and $m \in m_2 = \{0.1, 0.4, 0.7, \ldots, 2.2, 3, 5\}$ for τ_c^{mean}.

8.3.1 Conflict coloring is hard

Statistic. A combinatorial problem may display a phase transition [HW06], with instances moving from easy to hard when the parameters evolve. We study the hardness of conflict coloring in this respect, using the *fraction of valid configurations* $f_{c,m}$, namely fraction of valid configurations over the entire search space $\prod_{p_i \in P} C[p_i]$. We compute the value of $f_{c,m}$ (given the values of τ_c^{min} and τ_c^{mean}), by enumerating all these configurations. We repeat the calculation $R = 100$, and make a violin plot of the values obtained[1].

Results. The results show a marked difference for values of τ_c^{min} vs. τ_c^{mean} (Figure 8.3). For the former, values of $f_{1.01}$ and f_2 are very small, and overall, the distribution of $f_{c,m}$ is shifted towards smaller values when the number of colors c increases. This is sound, as increasing the number of colors is expected to decrease the intersection area, whence the value of $\tau_{\text{loc.}}^{\text{Max-Min}}$. On the other hand, $\tau_{\text{loc.}}^{\text{Max-Mean}}$ being defined from average values, one expect its value to be less sensitive to the number of colors (Figure 8.7). This lesser sensitivity is obvious from the violin plots associated with τ_c^{mean}, since these are much less sensitive to the number of colors.

8.3.2 Existence of one solution

Statistic. Our second analysis focuses on the existence of at least one solution, depending of our global thresholds τ_c^{min} and τ_c^{mean} (Equation 8.6). We run our algorithm on $R = 100$ instances, and define the success rate as $s_{c,m} = \#cases\ with\ one\ solution/R$.

Results. Statistics on $s_{c,m}$ complement those on $f_{c,m}$ Consider first the multiplicative factors m. It clearly appears that $s_{c,m}$ increases when increasing m, for both $\tau_{\text{loc.}}^{\text{Max-Mean}}$ and $\tau_{\text{loc.}}^{\text{Max-Min}}$ (Figure 8.4). Consider next the number of colors c. While the statistic $s_{c,m}$ hardly depends on the number of colors c for $\tau_{\text{loc.}}^{\text{Max-Mean}}$, it does so for $\tau_{\text{loc.}}^{\text{Max-Min}}$. Namely, in using $\tau_{\text{loc.}}^{\text{Max-Min}}$, $s_{c,m}$ decreases when increasing the number of colors. More precisely, our experiment shows that for $c \in \{5, 10, 15, 50, 100\}$, the average of $\tau_{\text{loc.}}^{\text{Max-mean}}$ is respectively around $(120, 100, 97, 90, 91)$ and $(38, 40, 11, 1.3, 0.05)$ for $\tau_{\text{loc.}}^{\text{Max-min}}$. It explains the statistics of $s_{c,m}$ (Figure 8.4) and also $f_{c,m}$ (Figure 8.3).

[1] To run the algorithm, we use Choco-a CSP solver see [JRL08].

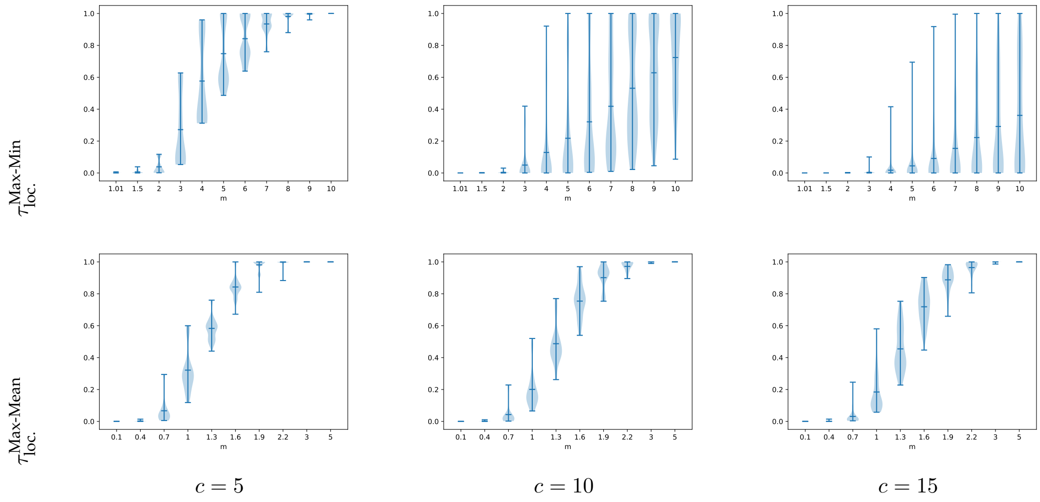
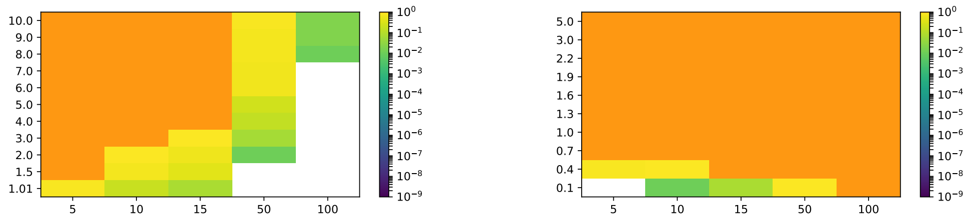


Figure 8.3: **Violin plots for the fraction $f_{c,m}$ of valid configurations in $\prod_{p_i \in P} C[p_i]$, for fixed values of the number of colors c and the multiplicative factors m in Equation 8.6.** Each plot features three sub-plots corresponding to the number of colors $c \in \{5, 10, 15\}$; each sub-plot features violins where each of which is corresponding to a multiplicative value in m_1 for for $\tau_{\text{loc.}}^{\text{Max-Min}}$, or m_2 for $\tau_{\text{loc.}}^{\text{Max-Mean}}$. Each violin plot corresponds to $R = 100$ random instances. These experiments involve $N' = 8$ finite Voronoi regions.



	5	10	15	50	100
1.01	0.85	0.18	0.07	0	0
1.5	1	0.68	0.4	0	0
2.0	1	0.9	0.6	0.01	0
3.0	1	1	0.87	0.06	0
4.0	1	1	1	0.15	0
5.0	1	1	1	0.22	0
6.0	1	1	1	0.64	0
7.0	1	1	1	0.66	0
8.0	1	1	1	0.69	0.01
9.0	1	1	1	0.71	0.02
10.0	1	1	1	0.8	0.02

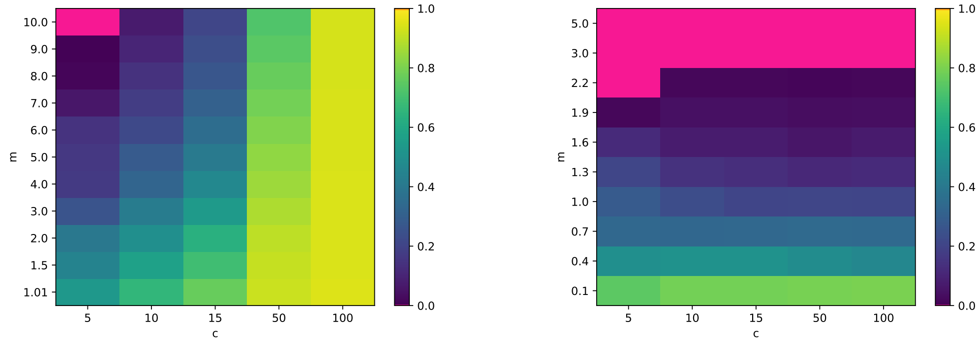
	5	10	15	50	100
0.1	0	0.01	0.07	0.86	1
0.4	0.82	0.89	1	1	1
0.7	1	1	1	1	1
1.0	1	1	1	1	1
1.3	1	1	1	1	1
1.6	1	1	1	1	1
1.9	1	1	1	1	1
2.2	1	1	1	1	1
3.0	1	1	1	1	1
5.0	1	1	1	1	1

$\tau_{\text{loc.}}^{\text{Max-Min}}$ $\tau_{\text{loc.}}^{\text{Max-Mean}}$

Figure 8.4: **Heat map representing the fraction $s_{c,m}$ of instances with at least one solution, for fixed values of the number of colors c (horizontal axis) and the multiplicative factor m (vertical axis).** In the heat-map and the Tables: rows (resp. columns) correspond to values of c (resp. m). A total of $R = 100$ random instances were used. White color infers value 0.

8.4 Running times

We provide the running times of checking the existence of one solution for $\tau_{\text{loc.}}^{\text{Max-min}}$ and $\tau_{\text{loc.}}^{\text{Max-mean}}$, see Figure 8.6. Clearly, the running augments when increasing c in general. Figure 8.6 also show that the time lines corresponding to values m in the



$\tau_{\text{loc.}}^{\text{Max-Min}}$ $\tau_{\text{loc.}}^{\text{Max-Mean}}$

Figure 8.5: **Heat map representing the fraction of conflict edges, averaged over all edges of the Delaunay triangulation. (Note that for a Delaunay edge, there are at most c^2 conflict edges.)** Total run of $R = 100$. Pink color maps to values in $[0, 10^{-3})$.

left plot are quite similar. In the other hand, those in the right plot tend to 0 when m increases which is to say that instances get easier. Hence, the difficulty of an instance does not depend on m (exactly with m chosen in our experiment) for $\tau_{\text{loc.}}^{\text{Max-min}}$ but it does for $\tau_{\text{loc.}}^{\text{Max-mean}}$.

On computational complexity point of view, Conflict Coloring is very hard which the running time for solving the problem is exponential to the size of the input. However, the statistics show that the running time does not explode for c big enough and N moderate. Indeed, they are less than 0.25 seconds for $N = 15$ (with $N' = 8$) and $c \leq 100$ (Figure 8.6). Hence, it is capable to solve the problem Conflict Coloring in practice (with instances of tens of subunits).

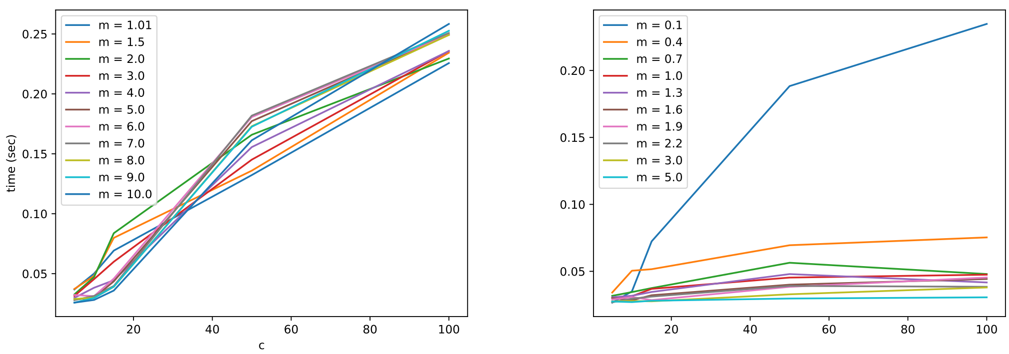


Figure 8.6: **Tables of average running times checking the existence of one solution for $\tau_{\text{loc.}}^{\text{Max-Min}}$ (left) and $\tau_{\text{loc.}}^{\text{Max-Mean}}$ (right) (in second).** Total run of $R = 100$.

8.5 Values and thresholds: upper bounds, lower bounds, and solutions

From our experimental work, the statistics provide useful information about the relation between local constraints and global solutions.

Application-wise, an important question is the adjustment of lower and upper values for the thresholds τ_c, in order to get useful information. In the sequel, we explain how to infer such values via the choice of the multiplicative factor m, for the case of $\tau_{\text{loc.}}^{\text{Max-mean}}$ – that of $\tau_{\text{loc.}}^{\text{Max-min}}$ is similar.

The upper bound is generally application dependent. Large values of τ_c are such that most colorings are valid, yet, such large τ_c may be irrelevant application-wise. For example, transposing our Delaunay-Voronoi setting to molecular modeling, large intersection volumes between atomic representations (the equivalent of the intersection area used in this work) directly translate into prohibitive van der Waals interaction energies [**?**] (which explains interactions of subunits from physical aspects).

To choose a lower bound, one seeks the smallest value of m yielding the desired value of $s_{c,m}$. In our experiments, for $s_{c,m} = 1$, $m \leq 0.7$ suffices whatever the value of c (the right plot of Figure 8.4). For $s_{c,m} = s_0(< 1)$, relevant pairs (c, m) are obtained by seeking the smallest values of m and c such that $s_{c,m} \geq s_0$.

In addition, we provide Figure 8.5 which shows the density of conflict graphs corresponding to pairs (c, m). In fact, the statistics of density is opposite to the one of $s_{c,m}$ (*i.e.* for a fixed c, when m increases, a conflict graph is clearly sparser and it is the contrary for a fixed m and when c increases). Regarding pairs (c, m) whose $s_{c,m} = 1$, the statistics show that the density is at most 47% when $(c = 15, m = 4)$ for $\tau_{\text{loc.}}^{\text{Max-min}}$ and at most 78% when $(c = 100, m = 0.1)$ for $\tau_{\text{loc.}}^{\text{Max-mean}}$. And the density expects to increase when c increases (but m does not) for $\tau_{\text{loc.}}^{\text{Max-min}}$ but it keeps stable for $\tau_{\text{loc.}}^{\text{Max-mean}}$ when c increases. In this latter case, a high fraction of conflict edges does not jeopardize the existence of solutions, as evidenced e.g. by the pair $(c = 100, m = 0.1)$ (fraction of conflict edges ~ 0.8 and $s_{100,0.1} = 1$).

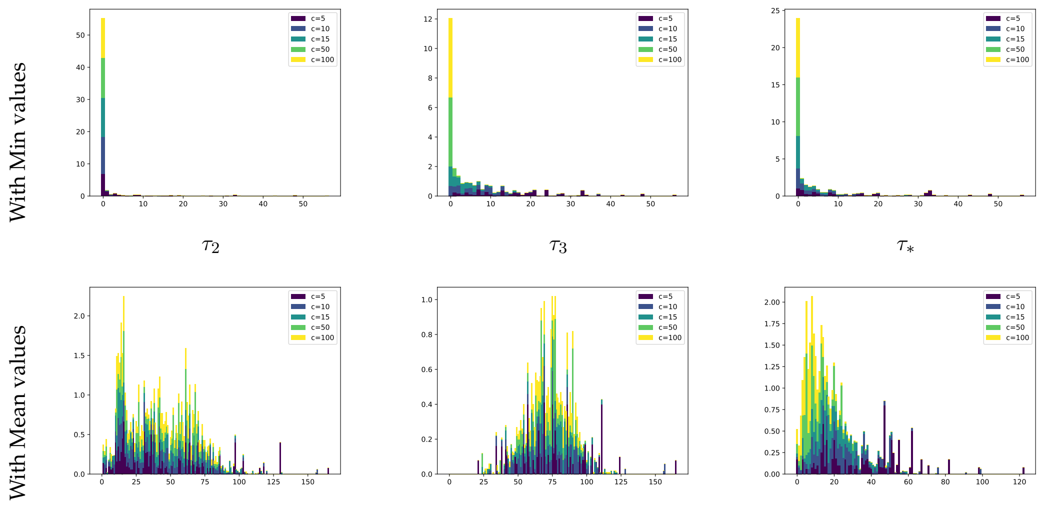


Figure 8.7: **Distribution of intersection areas $ia(\cdot,\cdot)$ for Min (top row) and Mean values (bottom row).** The three plots correspond to τ_2, τ_3 and τ_*, respectively. A total of $R = 100$ random instances were used. Note that y-axis presents the average number of edges, faces or stars of an instance; and x-axis infers values $\lfloor ia(\cdot,\cdot)\rfloor$.

Conclusion of Part II

In this part, we studied coloring problems based on DOMINO. We gave a complexity dichotomy for FULFILL COLORING, CONFLICT COLORING and their restriction to graph families $\mathcal{B}(k, m)$ of conflict k-graphs.

We have shown that the coloring problems are very hard in most graph families $\mathcal{B}(k, m)$. For a family $\mathcal{B}$ such that $\mathcal{B}$-FULFILL-k-COLORING is NP-complete and $\mathcal{B}$-CONFLICT-k-COLORING is in P, we can exploit the latter to build an algorithm for solving the former. Assume that $\mathcal{B}$ has the following property: *each color of one vertex typically pairs up with only one color of the other*, this yields the conflict graph which is a complete bipartite graph minus a matching. Let $\mathcal{A}$ the set of graphs obtained from $K_{k,k}$ by deleting a matching. One can show that $\mathcal{A}$-CONFLICT COLORING is polynomial-time solvable. Hence, although $\mathcal{A}$-FULFILL k-COLORING is NP-complete, when the number q of fulfilled edges is fixed, the problem can be solved in polynomial time. Indeed, for each of the $O(m^q)$ possible subgraphs F of G, one can test in polynomial time if F admits a k-coloring with no conflict edges.

In Section 7.1, we gave a 0.79607-approximation algorithm for FULFILL 2-COLORING. This approximation ratio is proved using an algorithm and analysis which are similar to the ones for MAX DICUT in [GW95]. Better SDP-based approximation algorithm for MAX DICUT have been shown. The current best one has an approximation ratio of 0.863 [MM01]. It would be interesting to investigate whether their technique can be generalized for FULFILL 2-COLORING. It is natural to study approximation algorithms for large value of k, that is to approximate FULFILL k-COLORING.

Further, more than finding a solution, it would be interesting to enumerate all possible solutions. Theoretically, it is NP-hard, then takes exponential time, so it would be useful to find moderately exponential algorithms.

Problem 92. Is there a constant c_k such that all conflict k-colorings of a graph can be enumerated in $O(f(n)c_k^n)$ time where n is the number of vertices and f is a polynomial function?

For the path P_n, it is easy to find a conflict coloring of P_n and we can even count the number of conflict colorings of P_n, both can be done in linear time. If the number of solutions is $O(c^n)$, then we can enumerate all solutions of P_n in $O(nc^n)$. Therefore, it can be helpful to start this problem with some simple graphs G.

Regarding the experimental part, our geometric setup is associated with Voronoi diagrams in 2-dimension. The statistics give several relations between considered parameters. It would be similar if the setup is 3-dimensional. Hence, our experiments allow us to estimate, in another setup, dependencies of parameters (which are used in our setup).

To structural biology, a central motivation for this work is the reconstruction of atomic models of macromolecular assemblies, based on (i) the interaction graph between subunits, and (ii) one set of conformations (colors) for each subunit. Two subunits in contact must exhibit a high steric complementarity, with in particular little overlap between atoms – that would otherwise, the interaction between two subunits becomes competitive and unbalanced. For this reason, the resulting conflict k-graphs are almost complete bipartite conflict k-graphs. We have noticed that in processing instances defined from the Delaunay-Voronoi setting, which mimics subunits with controlled overlap, a high fraction of conflict edges does not harm the existence of solutions. In our experimental settings, such a fraction can be at least 0.8 in average (*i.e.* 80% edges of the complete bipartite graph are in the conflict graph), and solutions are still found. It may be interesting to investigate further the frontier of density starting from which solutions cease to exist.

In practice, there are instances for which it is hard or even impossible to define the conflict graph of an edge. This is related to local properties of an input. For example, when all pairs of colors of an edge have similar weights which are much smaller compared to those of other edges, then choosing a global threshold may give complete conflict graphs for such edges. In this case, a possibility may be to use local thresholds, for example based on some local structural properties.

Moreover, an important issue in practice is to enumerate all sets of conformations for an assembly, because it presents different states of the assembly at times. Though this is very hard from a theoretical point of view, our experiments show that it is possible to enumerate all solutions for instances with moderate number of subunits.

Therefore, our further work would try to process cases with the order of tens of subunits, each exhibiting up to hundreds of conformations.

Bibliography

[AAR10] D. Angluin, J Aspnes, and L. Reyzin. Inferring social networks from outbreaks. In *ALT '10*, pages 104–118, 2010.

[ACCC15] D. Agarwal, C. Caillouet, D. Coudert, and F. Cazals. Unveiling Contacts within Macro-molecular assemblies by solving Minimum Weight Connectivity Inference Problems. *Molecular and Cellular Proteomics*, 14:2274–2284, 2015.

[AH77a] K. Appel and W. Haken. Every planar map is four colorable. Part I. Discharging. *Illinois J. Math.*, 21:429–490, 1977.

[AH77b] K. Appel and W. Haken. Every planar map is four colorable. Part II. Reducibility. *Illinois J. Math.*, 21:491–567, 1977.

[AH89] K. Appel and W. Haken. Every Planar Map Is Four Colorable. *A.M.S. Contemp. Math.*, 98, 1989.

[APT79] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8:121–123, 1979.

[BCM$^+$18] F. Bonomo, M. Chudnovsky, P. Maceli, O. Schaudt, M. Stein, and M. Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38(4), 2018.

[BCPS10] U. Brandes, S. Cornelsen, B. Pampel, and A. Sallaberry. Blocks of hypergraphs - applied to hypergraphs and outerplanarity. In *IWOCA '10*, pages 201–211, 2010.

[Ber57] C. Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43:842–844, 1957.

[Ber19] A. Bernshteyn. The Johansson-Molloy theorem for DP-coloring. *Random Structures and Algorithms*, 54(4):653–664, 2019.

[Bod93] H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceeding of STOC'1993*, pages 226–234, 1993.

[Bro41] R.L Brooks. On coloring the nodes of a network. *Mathematical Processing of the Cambridge Philosophical Society*, 37:194–197, 1841.

[Bul17] A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceeding of FOCS'2017*, pages 319–330, 2017.

[BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic geometry*. Cambridge University Press, 1998. Translated by H. Brönnimann.

[CDS04] V. Conitzer, J. Derryberry, and T. Sandholm. Combinatorial auctions with structured item graphs. In *Proceeding of AAAI'2004*, pages 212–218, 2004.

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CHM+17] N. Cohen, F. Havet, D. Mazauric, I. Sau, and R. Watrigant. Complexity dichotomies for the minimum F-overlay problem. In *Proceeding of IWOCA'2017)*, pages 116–127, 2017.

[CHM+19] N. Cohen, F. Havet, D. Mazauric, I. Sau, and R. Watrigant. Complexity dichotomies for the minimum F-overlay problem. *Journal of Discrete Algorithms*, 2019.

[CKN+15] J. J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchý, and M. Weller. Polynomial-time data reduction for the subset interconnection design problem. *SIAM J. Discrete Math.*, 29(1):1–25, 2015.

[CMTV07] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics. In *PODC '07*, 2007.

[Coo71] S.A. Cook. The complexity of theorem-proving procedures. *Proceeding of STOC'1971*, pages 151–158, 1971.

[CVFK+15] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. 2015.

[Dai80] D. P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics*, 30(3):289–293, 1980.

[DK95] D.-Z. Du and D. F. Kelley. On complexity of subset interconnection designs. *Journal of Global Optimization*, 6(2):193–205, 1995.

[DM88] D.-Z. Du and Z. Miller. Matroids and subset interconnection design. *SIAM Journal on Discrete Mathematics*, 1(4):416–424, 1988.

[DP17] Z. Dvořák and L. Postle. Correspondence coloring and its application to list-coloring planar graphs without cycles of lengths 4 to 8. *Journal of Combinatorial Theory, Series B*, 129, 09 2017.

[Edw86] K Edwards. The complexity of colouring problems on dense graphs. *Theor. Comput. Sci.*, 43, 1986.

[ES76] A. I. Even and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5:691–703, 1976.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[FHK16] P. Fraigniaud, M. Heinrich, and A. Kosowski. Local conflict coloring. In *Proceeding of FOCS'2016*, pages 625–634, 10 2016.

[FHWE08] H. Fan, C. Hundt, Y.-L. Wu, and J. Ernst. Algorithms and implementation for interconnection graph problem. In *Combinatorial Optimization and Applications*, pages 201–210, 2008.

[FV98] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and Constraint Satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[FW08] H. Fan and Y.-L. Wu. Interconnection graph problem. In *Proceedings of FCS'2008*, pages 51–55, 2008.

[GJPS17] P. Golovach, M. Johnson, D. Paulusma, and Jiangli Song. A survey on the computational complexity of colouring graphs with forbidden subgraphs. *J. Graph Theory*, 84(4):331–363, 2017.

[GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[Had75] F.O. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.*, 4:221–225, 1975.

[Hal35] P. Hall. On representatives of subsets. *J. London Math. Soc.*, 10:26–30, 1935.

[Hea90] P. J. Heawood. Map colour theorem. *Quart. J. Pure App1. Math.*, 24:332–338, 1890.

[HHI+12] J. Hosoda, J. Hromkovic, T. Izumi, H. Ono, M. Steinová, and K. Wada. On the approximability and hardness of minimum topic connected overlay and its special instances. *Theor. Comput. Sci.*, 429:144–154, 2012.

[HHKL90] K. Heinrich, P. Hell, D. G. Kirkpatrick, and G. Liu. A simple existence criterion for (g < f)- factors. *Discrete Mathematics*, 85(3):313–317, 1990.

[HK73] J. Hopcroft and R. Karp. An n5/2 algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[HKL+10] T.-C. Hoàng, M. Kamiński, V. Lozin, J. Sawada, and X. Shu. Deciding k-colorability of p5-free graphs in polynomial time. *Algorithmica*, 57, 2010.

[HMNW20] F. Havet, D. Mazauric, V.-H. Nguyen, and R. Watrigant. Overlaying a hypergraph with a graph with bounded maximum degree. In *Proceedings of CALDAM'2020*, volume 12016 of *LNCS*, pages 403–414, 2020.

[HN90] Pavol H. and Jaroslav N. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[Hol81] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981.

[HS78] E. Horowitz and S. Sahni. Fundamentals of computer algorithms. *Computer Science Press*, pages 151–158, 1978.

[HW06] A. K Hartmann and M. Weigt. *Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics*. John Wiley & Sons, 2006.

[JP87] D. S. Johnson and H. O. Pollak. Hypergraph planarity and the complexity of drawing Venn diagrams. *Journal of Graph Theory*, 11(3):309–325, 1987.

[JRL08] N. Jussien, G. Rochart, and X. Lorca. Choco: an open source java constraint programming library. In *Proceeding of OSSICP'08*, pages 1–10, 2008.

[Jun13] D. Jungnickel. *Graphs, Networks and Algorithms*, volume 5 of *Algorithms and Computation in Mathematics*. Springer-Verlag Berlin Heidelberg, 2013.

[Kar72] R. M. Karp. *Reducibility among Combinatorial Problems*. Springer US, 1972.

[KKW15] J. Köbler, S. Kuhnert, and O. Watanabe. Interval graph representation with given interval and intersection lengths. *J. Discrete Algorithms*, 34:108–117, 2015.

[KO19] S.-J. Kim and K. Ozeki. A note on a brooks' type theorem for DP-coloring. *Journal of Graph Theory*, 91(2):148–161, 2019.

[KS03] E. Korach and M. Stern. The clustering matroid and the optimal clustering tree. *Mathematical Programming*, 98(1):385–414, 2003.

[Lat94] R.H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP- complete. *Protein Engineering*, 7:1059–1068, 1994.

[Lic82] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982.

[LTSW09] K. Lasker, M. Topf, A. Sali, and H. Wolfson. Inferential optimization for simultaneous fitting of multiple components into a cryoem map of their assembly. *Journal of molecular biology*, 388(1):180–194, 2009.

[MM01] S. Matuura and T. Matsui. 0.863-approximation algorithm for MAX DICUT. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 138–146, 2001.

[OR11] M. Onus and A. Richa. Minimum maximum-degree publish-subscribe overlay network design. *IEEE/ACM Trans. Netw.*, 19(5):1331–1343, 2011.

[Rob03] S. Robinson. Are medical students meeting their (best possible) match? *SIAM News*, 36, 2003.

[RSST96] N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas. Efficiently four-coloring planar graphs. In *Proceedings of STOC'2016*, 1996.

[RSYG19] L. Runrun, L. Sarah, Y. Yuxue, and Y. Gexin. DP-3-coloring of some planar graphs. *Discrete Mathematics*, 342(1):178 – 189, 2019.

[Sch03] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003.

[Tar72] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.

[Tov84] A. C. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85 – 89, 1984.

[XJB05] J. Xu, F. Jiao, and B. Berger. A tree-decomposition approach to protein structure prediction. In *Proc IEEE Comput Syst Bioinform Conf*, 2005.

[XLKX03] J. Xu, M. Li, D. Kim, and Y. Xu. Raptor: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1:95–117, 2003.

[Zhu17] D. Zhuk. A proof of the CSP dichotomy conjecture. In *Proceeding of FOCS'2017*, pages 331–342, 2017.