Intersperse Coloring

by

Ehsan Chiniforooshan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis, we introduce the intersperse coloring problem, which is a generalized version of the hypergraph coloring problem. In the intersperse coloring problem, we seek a coloring that assigns at least ℓ different colors to each hyperedge of the input hypergraph, where ℓ is an input parameter of the problem.

We show that the notion of intersperse coloring unifies several well-known coloring problems, in addition to the conventional graph and hypergraph coloring problems, such as the strong coloring of hypergraphs, the star coloring problem, the problem of proper coloring of graph powers, the acyclic coloring problem, and the frugal coloring problem.

We also provide a number of upper and lower bounds on the intersperse coloring problem on hypergraphs in the general case. The nice thing about our general bounds is that they can be applied to all the coloring problems that are special cases of the intersperse coloring problem.

In this thesis, we also propose a new model for graph and hypergraph property testing, called the symmetric model. The symmetric model is the first model that can be used for developing property testing algorithms for non-uniform hypergraphs. We also prove that there exist graph properties that have efficient property testers in the symmetric model but do not have any efficient property tester in previously-known property testing models.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Naomi Nishimura. I owe her all my thanks for being encouraging and helping me to find the right method of research. This work would not have been possible without her invaluable help and guidance.

Also, I am thankful to my committee members, Therese Biedl, Jason Brown, Jochen Konemann, and Prabhakar Ragde for reading my thesis carefully and giving many valuable feedbacks. Additionally, I would like to thank my wife, Narges Simjour, for her support and for many useful discussions on this work I have had with her.

Finally, I would like to thank my parents, parents in law, sisters, and brothers in law for their constant encouragement and support.

To Narges

Contents

List of Figures

Chapter 1

Introduction

One of the reasons that the graph coloring problem has become one of the most interesting and popular combinatorial problems is its power to model other problems of different kinds. This means that, in many cases, solving other problems reduces to solving special cases of the graph coloring problem. For instance, the graph coloring problem can be used in scheduling, frequency assignment, register allocation, printed circuit board testing, pattern matching, and analysis of biological and archaeological data [78]. Our goal in this thesis is to generalize the graph coloring problem so that it can be used to model a wider range of problems.

Intuitively, the *graph coloring* problem is the problem of finding a color assignment for vertices of a given graph such that every edge of the graph is colored by two different colors (or simply is *bicolored*). Such a coloring is called a *proper coloring*. We want to find a proper coloring that uses the minimum number of colors. This minimum, for a graph A, is denoted by $\chi(A)$. A more general problem is the *hypergraph coloring* problem; *hypergraphs*, also known as *set systems*, are like graphs except that edges of a hypergraph, which are called *hyperedges*, can have more than two vertices. Again, in the hypergraph coloring problem we want to find a proper coloring, which is a color assignment for vertices of a given hypergraph such that every hyperedge is colored with at least two different colors. In the *intersperse coloring* problem, the generalized problem that we propose, we extend the condition of being bicolored to being colored by at least ℓ colors, where ℓ is a part of the input. If a hyperedge has a vertices, where $a < \ell$, we want it to be colored with a different colors, since it cannot get more than a colors. Such a coloring is called an *ℓ-intersperse coloring* and the minimum number of colors required in an ℓ-intersperse coloring of a hypergraph N is denoted by $\chi_\ell(N)$. Figure 1.1 illustrates a 3-intersperse coloring of a hypergraph N. In Figure 1.1, the hypergraph N has

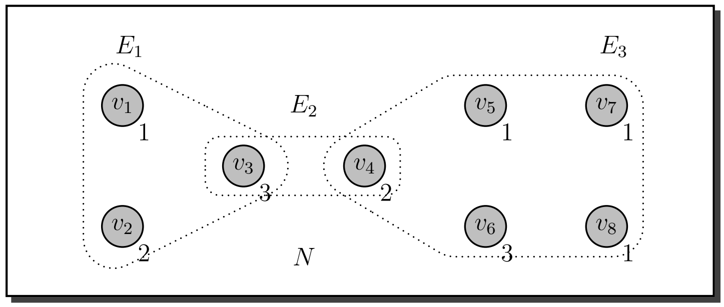

Figure 1.1: An example of an intersperse coloring where $\ell = 3$.

three hyperedges of sizes two, three, and five. Since the coloring, with colors 1, 2, and 3, shown in Figure 1.1 assigns three different colors to hyperedges E_1 and E_3 and two different colors to E_2, it is a 3-intersperse coloring.

In this thesis, in addition to studying the intersperse coloring problem on general hypergraphs, we also consider this problem on special families of hypergraphs. *Copy hypergraphs*, which are first defined in this thesis, are hypergraphs that are constructed based on a graph A and a family of graphs $\mathcal{B}$: the *$\mathcal{B}$-copy hypergraph of A* consists of the vertex set of A and a hyperedge for the vertices of every subgraph A' of A that is isomorphic to a graph in $\mathcal{B}$, where by *isomorphic* we mean A' can be made identical to a graph in $\mathcal{B}$ by renaming the vertices of A'. For example, in Figure 1.2, the $\{C_3, C_4\}$-copy hypergraph of A has three hyperedges, because A has two subgraphs that are isomorphic to C_3 and one subgraph that is isomorphic to C_4, where C_i is a cycle of length i.

For simplicity, we denote the problem of finding an ℓ-intersperse coloring for the $\mathcal{B}$-copy hypergraph of A with minimum number of colors by $\mathrm{SC}(A, \mathcal{B}, \ell)$, where SC stands for Subgraph Coloring. There are several coloring problems that can be viewed as special cases of $\mathrm{SC}(A, \mathcal{B}, \ell)$, and thus the intersperse coloring problem. Therefore, this thesis opens a new line of research by unifying several coloring problems. Below, we list a number of coloring problems that can be viewed as special cases of the intersperse coloring problem.

Perhaps, beside the graph and hypergraph coloring problems, the most straightforward special case of intersperse coloring is the *strong hypergraph coloring* problem [3], which is the problem of finding a coloring of vertices of a given hypergraph such that every two vertices that are in the same hyperedge get different colors. Hence, it can be viewed as an instance of the intersperse coloring problem in which ℓ is the size of the largest hyperedge.

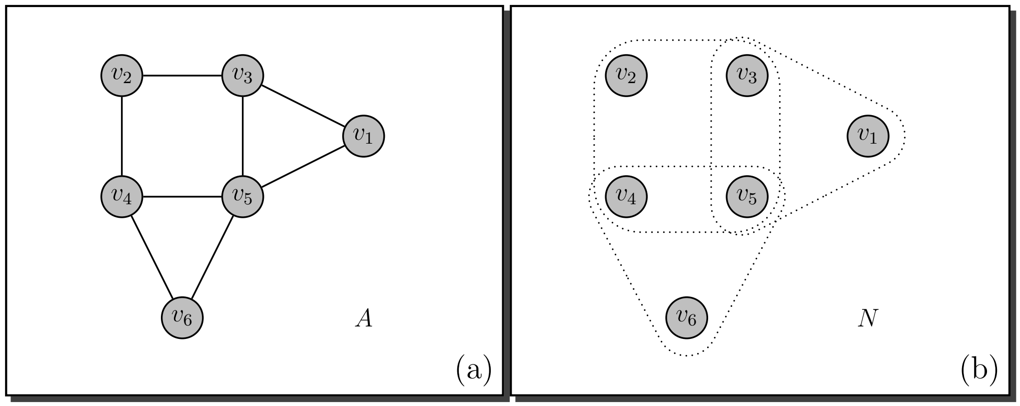

Figure 1.2: Part (b) illustrates the $\{C_3, C_4\}$-copy hypergraph of the graph in part (a).

Agnarsson and Halldórsson introduced the strong hypergraph coloring problem and developed online algorithms for this problem [3]. Also, they proved that every hypergraph N with m hyperedges can be strongly colored by at most $r\sqrt{m}$ colors, where r is the size of the largest hyperedge in N.

A special case of the intersperse coloring problem of copy hypergraphs is the problem of P *k-coloring* of graphs, where P is a family of graphs, i.e. a graph property, and k is an integer [21, 47]: in the P k-coloring problem one seeks a vertex coloring of the given graph such that the subgraph induced by each color class has graph property P. Brown and Corneil considered P k-coloring of graphs for the case in which P is the set of all graphs that have no induced subgraph isomorphic to any graph in $\mathcal{B}$. In other words, they want to find a vertex coloring of the input graph with k colors such that the subgraph induced by any color class does not have any subgraph isomorphic to a graph in $\mathcal{B}$. Therefore, the above-mentioned problem is equivalent to finding a 2-intersperse k-coloring of the $\mathcal{B}$-copy hypergraph of the input graph, i.e. $\mathrm{SC}(A, \mathcal{B}, 2)$.

Star coloring [38] is another example: in star coloring, first introduced by Fertin, Raspaud, and Reed [38], we need to find a proper coloring of a given graph A such that no path of length three in A is bicolored. Thus, the star coloring problem is equivalent to $\mathrm{SC}(A, \{P_1, P_3\}, 3)$, where P_i is a path of length i. Fertin et al. used $\chi_s(A)$ to denote the minimum number of colors required for this purpose and they called it the *star chromatic number* of A. Fertin et al. found the exact value of $\chi_s(A)$ for trees, cycles, complete bipartite graphs, and 2-dimensional grids [38]. Also, they provided bounds for star chromatic numbers for some other families of graphs.

The main technique for obtaining lower bounds in the work of Fertin et al. [38] is bounding the number of edges in the graph induced by any pair of colors. It can be observed that the graph induced by any pair of colors should be a forest, because

1. a cycle of length three cannot be properly colored with two colors, and
2. a cycle of length more than three has a path of length three, and thus, by the star coloring condition, cannot be properly colored with two colors.

Thus, the graph induced by any pair of colors cannot have many edges. Using this observation, they bound $\chi_s(A)$ from below. In Chapter 4, we generalize this method to obtain a lower bound for the intersperse chromatic number of copy hypergraphs. Fertin et al. also used a probabilistic method for proving an upper bound on the star chromatic number of bounded degree graphs. Hence, because of the probabilistic nature of the proof, their theorem is not constructive, i.e. they do not give a polynomial-time algorithm for finding the coloring itself. Also, Wood [84] investigated the star chromatic number of subdivisions of a general graph A: the subdivision of a graph is obtained by replacing each edge with a path of length two.

As another example, we consider proper coloring of the ith powers of graphs: the *ith power* of a graph A, denoted by A^i, has the same vertex set and for each u and v an edge between u and v is in A^i if and only if the distance between u and v in A is at most i. There are many papers on coloring graph powers [1, 2, 10, 66, 68, 82]. Coloring the ith powers of graphs is similar, but not equivalent, to a special case of $\mathrm{SC}(A, \mathcal{B}, \ell)$ where $\mathcal{B} = \{P_i\}$ and $\ell = i + 1$, because if we color the ith power of a graph properly, then every pair of vertices at distance at most i in the original graph is assigned two different colors. In particular, every pair of vertices on a path of length i in A is assigned two different colors in any proper coloring of A^i. Hence, if c is a proper coloring of A^i, then c assigns $i + 1$ different colors to the vertices of any path of length i in A. Thus, $\chi_{i+1}(N)$ is at most $\chi(A^i)$, where N is the $\{P_i\}$-copy hypergraph of A. However we can not say that $\chi(A^i)$ is always equal to $\chi_{i+1}(N)$. The reason is that $\mathrm{dist}(u, v) \leq i$ does not necessarily imply that u and v are contained in a path of length i. They may be in a shorter path. A counterexample is illustrated in Figure 1.3. A^3 is a complete graph and cannot be colored with fewer than six colors, but the four-coloring of A illustrated in Figure 1.3 assigns four colors to any path of length three, and hence, the P_3-copy hypergraph of A, N, has a 4-intersperse coloring with four colors. The problem of coloring the ith power of A is equivalent to $\mathrm{SC}(A, \{P_1, P_2, \ldots, P_i\}, i + 1)$, because if there

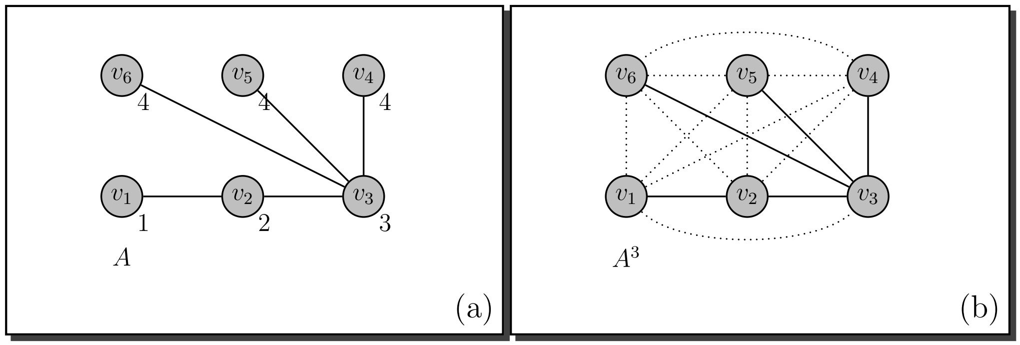

Figure 1.3: On the left a 4-intersperse coloring for the P_3-copy hypergraph of A is shown. The graph on the right is the third power of A.

is an edge between u and v in A^i, then u and v are in a path of length k, where $1 \leq k \leq i$, in A, and hence, a solution for $\mathrm{SC}(A, \{P_1, P_2, \ldots, P_i\}, i+1)$ will assign two different colors to u and v.

We start looking at previous results on the problem of coloring graph powers by considering a result of Alon and Mohar [10]. Alon and Mohar have shown that for graphs A with girth less than seven, $\chi(A^2)$ can be as large as $(1+o(1))\Delta^2(A)$, where the *girth* of A is the length of shortest cycle in A, and $\Delta(A)$ is the maximum number of edges that share a single vertex in A. They have also shown that for graphs with girth at least $3k+1$ the value of $\chi(A^k)$ is $\theta(\frac{\Delta^2(A)}{\log \Delta(A)})$ [10]. The complicated part of their result is obtaining the lower bound for $\chi(A^2)$. For this purpose, they use a probabilistic argument to show that there exists a graph A with large girth and low degree such that A^2 contains no independent set of size $\Omega(\frac{n}{\Delta^2(A)} \log \Delta(A))$. Since in any proper coloring the set of vertices with the same color forms an independent set, the chromatic number of A^2 must be $\Omega(\frac{\Delta^2(A)}{\log \Delta(A)})$.

For planar graphs, Agnarsson and Halldórsson have proved that $\chi(A^k)$ is $O(\Delta^{\lfloor k/2 \rfloor}(A))$. They have also proposed a 2-approximation scheme for finding $\chi(A^2)$ where A is a planar graph [2]. They introduce a technique called the *contraction technique* to solve the problem of coloring squares of planar graphs. Contracting an edge in A, intuitively, is the process of putting the two ends of the edge $E \in \mathcal{E}(A)$ on each other to make them one vertex; the resulting graph is denoted by A/E. Therefore, after one edge contraction the number of vertices of the graph decreases by one. By contracting an edge in a planar graph we will obtain another planar graph. In the method of Agnarsson et al. [2] they find two neighbors $u \in V(A)$ and $v \in V(A)$ such that their contraction does not increase the maximum degree of the graph, and then they solve the problem for the smaller graph $A/\{u, v\}$.

Such an edge always exists in maximal planar graphs with $\Delta \geq 11$; a *maximal planar graph* is a planar graph A such that no edge can be added to A without violating the planarity. For planar graphs that are not maximal, they find a maximal supergraph and use the technique on the supergraph. We refer the interested reader to the paper by Agnarsson and Halldórsson [2] for more details. Molloy and Salavatipour improved this result and obtained a 5/3-approximation scheme for finding the chromatic number of the square of a planar graph [68].

Our fifth example is the problem of acyclic coloring [45]; *acyclic coloring* of a graph A is a proper coloring of A such that the subgraph induced by any two color classes is acyclic. In other words, any cycle of A gets at least three different colors. The minimum number of colors needed in an acyclic coloring of a graph is called its *acyclic chromatic number*. Hence, the acyclic coloring problem is equivalent to $\mathrm{SC}(A, \mathcal{B}, 3)$, where $\mathcal{B} = \left\{P_2, C_3, C_4, \ldots, C_{|V(A)|}\right\}$ and C_i is a cycle of length i.

Acyclic coloring has been largely studied since 1973 and the acyclic chromatic number of several families of graphs such as planar graphs [20], d-dimensional grids [36], graphs of maximum degree three [45], and graphs of maximum degree four [23] has been determined. For general graphs, Fertin and Raspaud showed that there is a polynomial-time algorithm that, for any given graph A, can find an acyclic coloring of A with at most $\frac{\Delta(A)(\Delta(A)-1)}{2}$ colors [37].

The problem of frugal coloring [50] can be viewed as a special case of $\mathrm{SC}(A, \mathcal{B}, \ell)$; a *$\beta$-frugal coloring* of a given graph is a proper coloring such that at most β neighbors of each vertex are of the same color. Frugal coloring is useful in total coloring, in which the goal is to find a color assignment to vertices and edges of a graph so that adjacent vertices, edges that have a vertex in common, and an edge and its two end-vertices have different colors. Also, frugal coloring is used in channel-allocation schemes in multi-channel multi-radio wireless networks [51, 71].

To see the relation between the frugal coloring problem and the intersperse coloring problem, note that a frugal coloring of A assigns at least three colors to any subgraph of A isomorphic to $S_{\beta+1}$, where $S_{\beta+1}$ is the star graph on $\beta+2$ vertices, shown in Figure 1.4. Also, any proper coloring that assigns at least three colors to any subgraph of A isomorphic to $S_{\beta+1}$ is a frugal coloring of A. Consequently, the problem of finding a β-frugal coloring with the minimum number of colors for A is equivalent to $\mathrm{SC}(A, \{P_1, S_{\beta+1}\}, 3)$.

For general graphs, Hind et al. proved that any graph A has an $O(\log^5 \Delta(A))$-frugal coloring with $\Delta(A) + 1$ colors and one such coloring can be found in polynomial time [51]. Pemmaraju and Srinivasan improved this result and showed that A has an $O(\frac{\log^2 \Delta(A)}{\log \log \Delta(A)})$-frugal coloring with $\Delta(A) + 1$ colors [71]. For a triangle-free

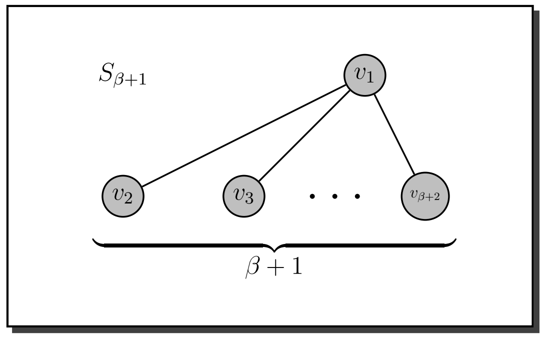

Figure 1.4: A star graph on $\beta+2$ vertices.

graph A, Pemmaraju and Srivinasan proved that A has an $O(\log^2\Delta(A))$-frugal coloring with $O(\frac{\Delta(A)}{\log\Delta(A)})$ colors [71].

Probably, the most important contributions of this thesis are introducing the intersperse coloring problem on general hypergraphs and on copy hypergraphs. As mentioned above, the intersperse coloring problem on copy hypergraphs has several well-known coloring problems as its special cases. Thus, the problem we define in this thesis unifies many coloring problems. In this thesis, our main goal is to obtain results that work for all instances of the intersperse coloring problem. However, we also consider two special cases in Subsection 4.2.2 and Chapter 5. The precise problem definition and related terminology are presented in Chapter 2.

In addition to introducing the above-mentioned problems, our other contribution is to provide a number of general upper and lower bounds on the intersperse coloring problem on hypergraphs and copy hypergraphs. Our general bounds can be applied to any coloring problem that is a special case of the intersperse coloring problem. For example, Theorem 3.1.2, which works for any instance of the intersperse coloring problem, gives a better upper bound for the strong coloring problem than the upper bound of $r\sqrt{m}$ by Agnarsson and Halldórsson in the special case of hypergraphs that have few large hyperedges. In Chapter 3 we will show how our result can be applied to the strong coloring problem in more detail. To prove the main theorem of Chapter 3, we use probabilistic methods and give an existential proof for an upper bound on the minimum number of colors needed in an intersperse coloring of a general hypergraph. We show that this theorem "almost" matches the known results on proper graph and hypergraph coloring. Then, we derandomize our method and give a polynomial-time algorithm to find such a coloring.

Chapter 4 is devoted to studying the intersperse coloring problem of copy hypergraphs. We obtain upper bounds and a lower bound for general copy hypergraphs.

Then, as a special case of our problem, we investigate $\mathrm{SC}(A, \{P_2\}, 2)$, where A is a graph. For this case, we investigate the property testing framework, and propose a revised model, called the *symmetric model*. We prove that checking whether a graph has a path of length two as an induced subgraph can be done with constant number of queries.

In Chapter 5 we study the proper hypergraph coloring problem for a special family of hypergraphs, called "geometric hypergraphs", introduced by Smorodinsky [76]. We obtain the first non-trivial bounds for a conjecture proposed by Smorodinsky [76] regarding the chromatic number of a family of geometric hypergraphs.

Finally, in Chapter 6 we briefly summarize our results and propose a number of open questions.

Chapter 2

Preliminaries

In this chapter, we introduce the basic notation, definitions, and terminology needed for reading this thesis.

2.1 Hypergraphs

Hypergraph theory is an important area in discrete mathematics and computer science; a pair $N = (V, \mathcal{E})$ is called a *hypergraph* where V is a finite set and $\mathcal{E} \subseteq 2^V$ is a family of subsets of V. The *rank* of a hypergraph N is r if the largest set $E \in \mathcal{E}$ is of size r. We call N a k*-uniform* hypergraph if every set $E \in \mathcal{E}$ is of size k. Also, a hypergraph N is called *uniform* if it is k-uniform for some integer k. A 2-uniform hypergraph is called a *graph*. Each element in V (resp. $\mathcal{E}$) is called a *vertex* (resp. a *hyperedge*).

Throughout this thesis, all hyperedges are of size more than one. We call a hyperedge of size two an *edge*. We use $V(N)$ and $\mathcal{E}(N)$ to denote the set of vertices and the set of edges of hypergraph N. For every vertex $v \in V$, the *degree* of v is defined as the number of hyperedges that include v, and is denoted by $d_N(v)$. Also, we use $\delta(N)$ and $\Delta(N)$ to denote the minimum degree and the maximum degree among the vertices of N. A family of hypergraphs is called *bounded degree* if there exists an integer k such that $\Delta(N)$ is at most k for any hypergraph N in that family.

Hypergraphs are one of the most general structures on a set. A hypergraph N can also induce some substructures on a subset of $V(N)$. The following are the definitions of two kinds of substructures of a hypergraph.

DEFINITION 2.1.1. Suppose $N = (V, \mathcal{E})$ and $N' = (V', \mathcal{E}')$ are two hypergraphs (graphs). Then, N' is called a *subhypergraph* (*subgraph*) of N if $V' \subseteq V$ and $\mathcal{E}' \subseteq \mathcal{E}$.

DEFINITION 2.1.2. Suppose $N = (V, \mathcal{E})$ is a hypergraph. If U is a subset of V, then $(U, \{E \cap U : E \in \mathcal{E}, |E \cap U| > 1\})$ is called the hypergraph *induced* by U and is denoted by $N[U]$. Also, the hypergraph *strictly induced* by U is defined as $(U, \{E : E \in \mathcal{E}, E \subseteq U\})$ and is denoted by $N[\![U]\!]$.

For more information on hypergraphs we refer the reader to a book by Berge on this topic [18].

There are numerous problems that can be considered on hypergraphs. The next section is devoted to defining the hypergraph problems that we consider in this thesis.

2.2 Problem Definition

In this thesis, we study the problem of finding proper colorings for hypergraphs.

DEFINITION 2.2.1. A *vertex coloring* (or simply a *coloring*) *of* N *with* k *colors* is a mapping c from $V(N)$ to $\{1, 2, \cdots, k\}$. If all vertices of a hyperedge of N are assigned to the same number, then that hyperedge is said to be *monocolored*. Moreover, a hyperedge whose nodes have exactly two colors is called *bicolored*.

DEFINITION 2.2.2. A coloring c of a hypergraph N is a *proper vertex coloring* (or simply a *proper coloring*) if there is no monocolored hyperedge in N. We call N k*-colorable* if it has a proper coloring with k colors. The minimum value of k for which N is k-colorable is called the *chromatic number* of N and is denoted by $\chi(N)$. Furthermore, for any coloring c of N, we say the hyperedge $E \in \mathcal{E}(N)$ is *properly colored* if E is not monocolored.

Note that any proper coloring c of N with k colors partitions $V(N)$ into k disjoint sets $C_1, C_2, \ldots, C_k$, where C_i is the set of vertices of N that are colored with i. We call $C_1, C_2, \ldots, C_k$ the *color classes* of c. Since c is a proper coloring of N, there is no hyperedge $E \in \mathcal{E}(N)$ such that $E \subseteq C_i$ for some $1 \leq i \leq k$. Subsets of $V(N)$ with this property are called independent sets; more precisely, a subset U of $V(N)$ is an *independent set* if and only if there is no hyperedge $E \in \mathcal{E}(N)$ such that $E \subseteq U$. Thus, c can be viewed as a partitioning of $V(N)$ into k disjoint independent sets.

One method for proving a hypergraph N is k-colorable is to show that the minimum degree of all its strictly induced hypergraphs are at most $k-1$. We call such a hypergraph a $(k-1)$-*degenerate hypergraph*. Using a simple proof by induction, one can prove that a $(k-1)$-degenerate hypergraph is k-colorable [83].

LEMMA 2.2.3. *([83]) Every $(k-1)$-degenerate hypergraph is k-colorable.*

Proof. If there is no vertex in the hypergraph, it is obviously k-colorable; otherwise, the hypergraph has a vertex v with degree at most $k-1$. We eliminate v and hyperedges containing it, color the smaller hypergraph with k colors, then add v and hyperedges containing v, and assign a color from $\{1, 2, \ldots, k\}$ to v such that all hyperedges containing v have at least two different colors. It is possible because there are at most $k-1$ hyperedges containing v. □

The method of coloring the vertices that was used in the proof of Lemma 2.2.3 is called *the greedy coloring algorithm*. We will generalize the greedy coloring algorithm in Chapter 4. The following is a direct result of Lemma 2.2.3 [83].

COROLLARY 2.2.4. *([83]) For every hypergraph N, we have $\chi(N) \leq \Delta(N) + 1$.*

Proof. For every $U \subseteq V(N)$, the maximum degree of $N[\![U]\!]$ is at most $\Delta(N)$. Therefore, N is $\Delta(N)$-degenerate, and hence, according to Lemma 2.2.3, N is $\Delta(N) + 1$ colorable. □

In this thesis, we propose a generalized version of Definition 2.2.2:

DEFINITION 2.2.5. A coloring c of a hypergraph $N = (V, \mathcal{E})$ is an *ℓ-intersperse coloring* if every hyperedge $E \in \mathcal{E}$ is *ℓ-full colored*, that is, E is assigned at least $\min\{\ell, |E|\}$ different colors. We call N *ℓ-intersperse k-colorable* if it has an ℓ-intersperse coloring with k colors. The minimum value of k for which N is ℓ-intersperse k-colorable is denoted by $\chi_\ell(N)$ and is called the *ℓ-intersperse chromatic number* of N.

Note that the problem of finding an ℓ-intersperse coloring is a generalization of the problem of finding a proper coloring because $\chi(N) = \chi_2(N)$. We define the *intersperse coloring problem* as the problem of finding an ℓ-intersperse coloring for N, when N and ℓ are input parameters, that uses $\chi_\ell(N)$ colors. In the rest of this chapter we discuss different ways of attacking this problem.

2.3 Classes of Hypergraphs

As will be discussed in more detail in Chapter 3, finding an ℓ-intersperse coloring for a given hypergraph N that uses as few colors as possible is hard from the algorithmic point of view. Thus, it is natural to consider special families of hypergraphs. We have already defined two families of hypergraphs: bounded-degree hypergraphs and uniform hypergraphs. In this section we define more families of hypergraphs that will be examined in this thesis. However, we first, briefly, define a number of classes of graphs that will be used. For more details on the following definitions please refer to a standard textbook on graphs, such as the one by West [83].

Before we list a number of hypergraph families, we define the following two graph families, that will be used many times through out this thesis.

An n-vertex graph A is a *complete graph*, denoted by K_n, if $\{u, v\} \in \mathcal{E}(A)$ for all distinct vertices $u, v \in V(A)$. Moreover, a complete subgraph of a graph is called a *clique*.

An n-vertex *cycle*, denoted by C_n, is an n-vertex graph A that has n edges $\mathcal{E}(A) = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_n, v_1\}\}$, where $\{v_1, v_2, \ldots, v_n\}$ is the set of vertices of A. A graph A has a cycle if it has a subgraph isomorphic to a cycle.

2.3.1 Acyclic Hypergraphs

Probably one of the simplest families of graphs is the family of acyclic graphs defined below:

Definition 2.3.1. An *acyclic graph* is a graph that has no cycles. A *tree* is a connected acyclic graph.

The following is a well-known property of acyclic graphs [83].

Observation 2.3.2. *Every acyclic graph with at least one edge has a degree-one vertex.*

Since every subgraph of an acyclic graph is again an acyclic graph, we can repeatedly remove a degree-one vertex and the edge containing it from an acyclic graph until we are left with a graph with no edges. On the other hand, if a graph has a cycle, then the edges of the cycle will never be removed if we remove only degree-one vertices. Hence, another characterization of acyclic graphs is the following.

Observation 2.3.3. *A graph A is acyclic if and only if after removing degree-one vertices from A, until no more degree-one vertex removal is possible, A has no edges.*

Kuper [61] used the characterization of acyclic graphs stated in Observation 2.3.3 to define acyclic hypergraphs:

Definition 2.3.4. ([16]) A vertex v of a hypergraph N is called an *isolated* vertex if it is in only one hyperedge. The *Graham reduction* of N is obtained by doing the following steps, in any order, until it is no longer possible:

1. Delete an isolated vertex $v \in V(N)$.

2. If $E_1 \subseteq E_2$ for two hyperedges $E_1, E_2 \in \mathcal{E}(N)$, delete E_1.

Finally, N is called an *acyclic hypergraph* if its Graham reduction has at most one hyperedge.

Note that, if N is a graph, isolated vertices are exactly leaves of the graph. Note that the Graham reduction of a hypergraph is unique, because

1. an isolated vertex will remain an isolated vertex, and

2. if $E_1 \subseteq E_2$, then E_1 will remain a subset of E_2 until we remove E_1.

Examples of an acyclic hypergraph and a non-acyclic hypergraph are shown in Figure 2.1. The hypergraph N in Figure 2.1 is acyclic because we can remove vertices and hyperedges in the order $\langle v_3, v_2, E_1, v_1, E_2, v_4, E_3 \rangle$. Acyclic hypergraphs have many other definitions that are equivalent to Definition 2.3.4 [17]. They play an important role in the design of database schemes [16, 17, 64, 65].

Acyclic graphs are known to be 2-colorable [83]. This can be generalized to uniform acyclic hypergraphs.

Theorem 2.3.5. *Suppose N is an r-uniform acyclic hypergraph and $1 \leq \ell \leq r$ is an integer. Then, there is an ℓ-intersperse ℓ-coloring for N.*

Proof. We use induction on the number of hyperedges plus the number of vertices: if N has at most one hyperedge, then the theorem is trivial.

Assume that the theorem is true for all uniform acyclic hypergraphs whose number of vertices plus the number hyperedges is at most k, where $k \geq 1$. Also, suppose that N is an r-uniform acyclic hypergraph and $|V(N)| + |\mathcal{E}(N)| = k + 1$.

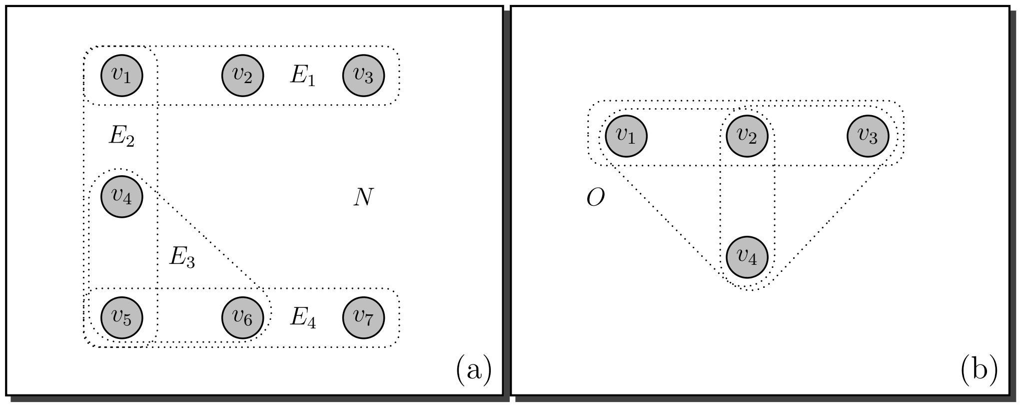

Figure 2.1: The hypergraph N on the left is an example of an acyclic hypergraph, while the hypergraph O on the right is not acyclic.

If there is a hyperedge $E_1 \in \mathcal{E}(N)$ that is a subset of another hyperedge, we can remove E_1 and we are done by induction. Otherwise, since N is acyclic, we know that by removing all isolated vertices of N we will find a hyperedge E that is a subset of another hyperedge F of N. Therefore, all vertices of $E - F$ are isolated vertices. The hypergraph $N' = (V(N) - (E - F), \mathcal{E}(N) - \{E\})$ is r-uniform, acyclic, and has $|V(N')| + |\mathcal{E}(N')| \leq k$. Thus, N' has an ℓ-intersperse ℓ-coloring c. Since the vertices of $E - F$ do not appear in any hyperedge in $\mathcal{E}(N) - \{E\}$ and so, no matter how we alter the color of vertices of $E - F$, c will remain an ℓ-intersperse coloring for N'. Since N is r-uniform, $|E - F| = |F - E|$; thus, we can assume $u_1, u_2, \ldots, u_d$ and $v_1, v_2, \ldots, v_d$ are the vertices of $E - F$ and $F - E$ respectively, where $d = |E - F| = |F - E|$. We define a new coloring c' which is equal to c on every vertex of N except the vertices in $E - F$. For the vertices in $E - F$ we set $c'(u_i) = c(v_i)$. Consequently, c' will be an ℓ-intersperse coloring for N', and the set of colors assigned to E by c' will be exactly the set of colors assigned to F. This means that c' is an ℓ-intersperse ℓ-coloring for N. $\square$

The special case in which N is 2-uniform and $\ell = 2$ is the above-mentioned 2-colorability property of acyclic graphs.

2.3.2 Neighborhood Hypergraphs

In this section, we consider a family of hypergraphs that is used to model the "neighborhood relations" in graphs. In general, a hypergraph can be used to represent certain substructures of a given graph: for each subset of the vertices of the

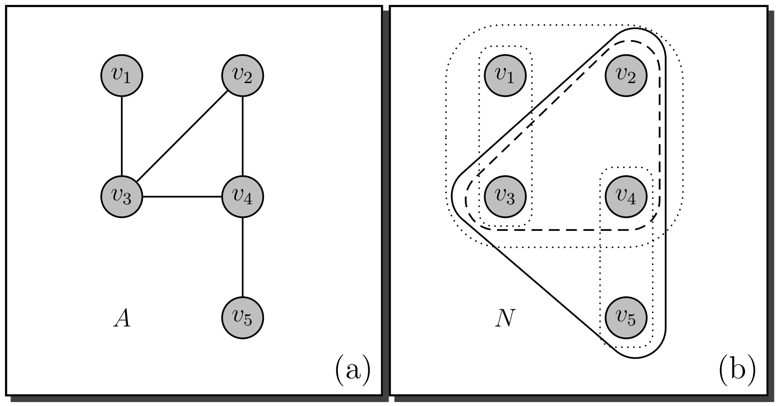

Figure 2.2: A graph A with its neighborhood hypergraph N.

input graph that has a specified structure or property, we build a hyperedge. The following is a detailed example.

DEFINITION 2.3.6. Suppose $A = (V, \mathcal{E})$ is a graph and $v \in V$ is a vertex of A. The *neighborhood* of v, denoted by $\Gamma_A(v)$, is the set of all vertices that have an edge to v, i.e. $\Gamma_A(v) = \{u \in V : \{u, v\} \in \mathcal{E}\}$. The *closed neighborhood* of v, denoted by $\Gamma^+{}_A(v)$, is defined as $\Gamma_A(v) \cup \{v\}$. The *neighborhood hypergraph* of A is the hypergraph $N = (V, \mathcal{E}')$ with vertices V and hyperedges $\mathcal{E}' = \{\Gamma^+{}_A(v) : v \in V\}$.

Figure 2.2 illustrates a graph together with its neighborhood hypergraph.

Definition 2.3.6, or concepts similar to Definition 2.3.6, are considered by different researchers. To the best of our knowledge, the above definition was introduced by Chastel et al. for the first time [24]. They studied the problem of coloring hyperedges of a given neighborhood hypergraph. We refer the reader to their paper [24] for more details.

As another example, see the paper by Fomin et al. [39]. Fomin et al. used neighborhood hypergraphs to develop an algorithm for finding the "domatic number", which is defined below, of the input graph [40, 39].

DEFINITION 2.3.7. Suppose that $A = (V, \mathcal{E})$ is a graph. A *dominating set* in A is a subset D of V such that every vertex $v \in V - D$ has a neighbor in D. In other words, for every vertex $v \in V$, $D \cap \Gamma^+{}_A(v)$ is not empty.

The *domatic number* of A is the maximum number k such that V can be partitioned into k disjoint dominating sets.

By an easy argument, we show that finding the domatic number of a graph A is a special case of the problem of computing $\chi_\ell(N)$, where ℓ is an integer and N is a neighborhood hypergraph. Assume that the neighborhood hypergraph of A is k-intersperse k-colorable, where $k \leq \delta(A)+1$. Then, each color class is a dominating set in A. Therefore, the domatic number of A is at least k. On the other hand, if the domatic number of A is at least k, then there exists a partitioning of $V(A)$ into k disjoint dominating sets $D_1, D_2, \ldots, D_k$. Therefore, if we assign the vertices of D_i to the ith color, each closed neighborhood ${\Gamma^+}_A(v)$ will have at least one color from each of the k color classes, for every $v \in V(A)$. Thus, the neighborhood hypergraph of A is k-intersperse k-colorable. So, we have the following property.

Observation 2.3.8. *The domatic number of a graph A is at least k if and only if $k \leq \delta(A)+1$ and the neighborhood hypergraph of A is k-intersperse k-colorable.* □

One can generalize the concept of neighborhood and obtain the following definition.

Definition 2.3.9. Suppose $A = (V, \mathcal{E})$ is a graph and u and v are two distinct vertices of A. A *path of length* n between u and v is a sequence of vertices $a_0, a_1, \ldots, a_n$ such that $a_0 = u$, $a_n = v$, $a_i \neq a_j$ if $i \neq j$, and for all $0 \leq i < n$ $\{a_i, a_{i+1}\} \in \mathcal{E}$. The *distance* between u and v in A is n if there is a path of length n between u and v and there is no path of length less than n between them. We denote the distance between u and v in A by $\mathrm{dist}(u, v)$. The *neighborhood of radius* r of v, denoted by $\Gamma_A(v, r)$, is the set of all vertices of distance at most r to v. The *closed neighborhood of radius* r of v, denoted by ${\Gamma^+}_A(v, r)$, is defined as $\Gamma_A(v, r) \cup \{v\}$. The *neighborhood hypergraph of radius* r of A is the hypergraph $N = (v, \mathcal{E}')$ with vertices v and hyperedges $\mathcal{E}' = \{{\Gamma^+}_A(v, r) : v \in V\}$.

A similar generalization of a dominating set of a graph A leads to the concept of r-distance dominating sets: an r-distance dominating set is a subset D of $V(A)$ such that ${\Gamma^+}_A(v, r) \cap D$ is not empty, for every $v \in V(A)$ [49]. Naturally, the *r-distance domatic number* of a graph A can be defined as the maximum number k such that A has k disjoint r-distance dominating sets. Finding distance dominating sets and the distance domatic number of graph have applications in networks [35], facility location problems [74], and other areas. We refer the reader to work by Simjour [74] for more details on the problems related to the distance domatic number problem.

Analogous to Observation 2.3.8, it can be shown that finding the r-distance domatic number of a graph is a special case of finding the maximum number of colors k such that the neighborhood hypergraph of radius r of the input graph has a k-intersperse k-coloring.

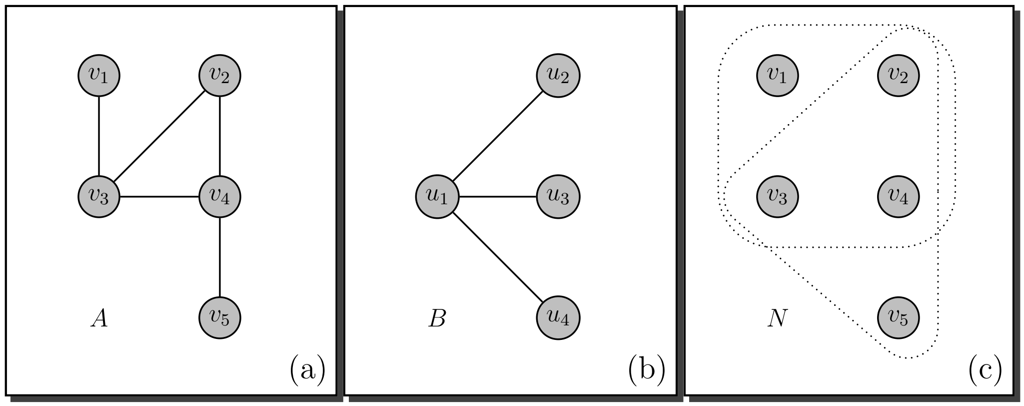

Figure 2.3: Two graphs A and B, and the B-copy hypergraph of A, N.

2.3.3 Copy Hypergraphs

In this subsection, we introduce another family of hypergraphs which are similar to neighborhood hypergraphs in the sense that they both represent certain substructures of a graph. For a given graph A, we can build a hypergraph in this way: we look at every subset U of $V(A)$ and we add U as a hyperedge if $A[U]$ has a subset of a "special shape". In order to define what we mean by a "special shape" we need the following definition.

DEFINITION 2.3.10. Suppose that N and O are two hypergraphs. An *isomorphism* between N and O is a bijection $c : V(N) \mapsto V(O)$ such that $E \in \mathcal{E}(N)$ if and only if $\{c(u) : u \in E\} \in \mathcal{E}(O)$. N and O are called *isomorphic* if there exists an isomorphism between N and O.

Intuitively, two hypergraphs (graphs) are isomorphic if we can obtain one of them from the other one just by renaming its vertices.

DEFINITION 2.3.11. The *B-copy hypergraph* of A, where A is a graph and B is a connected graph, is the hypergraph with vertices $V(A)$ and hyperedges $\mathcal{E} = \{U \subseteq V(A) : B \text{ is isomorphic to a spanning subgraph of } A[U]\}$. Similarly, the B-*induced copy hypergraph* of A is the hypergraph with vertices $V(A)$ and hyperedges $\mathcal{E} = \{U \subseteq V(A) : B \text{ is isomorphic to } A[U]\}$.

So, intuitively, the B-copy hypergraph of A has a hyperedge for every subgraph of A that has a copy of B in it.

Figure 2.3 shows an example of a copy hypergraph.

The following is a generalization of Definition 2.3.11.

DEFINITION 2.3.12. The *$\mathcal{B}$-copy hypergraph* of A, where A is a graph and $\mathcal{B}$ is a family of connected graphs, is the hypergraph with vertices $V(A)$ and hyperedges $\mathcal{E} = \{U \subseteq V(A) : \exists B \in \mathcal{B}$ such that B is isomorphic to a spanning subgraph of $A[U]\}$. Similarly, the *$\mathcal{B}$-induced copy hypergraph* of A is the hypergraph with vertices $V(A)$ and hyperedges $\mathcal{E} = \{U \subseteq V(A) : \exists B \in \mathcal{B}$ such that B is isomorphic to $A[U]\}$.

So, if $\mathcal{B}$ contains only one graph B, the $\mathcal{B}$-copy hypergraph of A is equal to the B-copy hypergraph of A. The same argument is true for $\mathcal{B}$-induced-copy hypergraph and B-induced-copy hypergraph. Thus, from now on, we only consider Definition 2.3.12.

Note that if the input contains A and $\mathcal{B}$, and not the copy hypergraph itself, there exists no polynomial-time algorithm for computing the $\mathcal{B}$-copy hypergraph or the $\mathcal{B}$-induced-copy hypergraph of A, unless $\mathcal{P} = \mathcal{NP}$. If $\mathcal{B}$ consists of only one graph, the above-mentioned problem is known as the *subgraph isomorphism problem* [42]. Hence, the $\mathcal{NP}$-hardness of the subgraph isomorphism problem [42] is also valid for the problem of computing the $\mathcal{B}$-copy hypergraph or the $\mathcal{B}$-induced copy hypergraph of A. In particular, note that the $\{P_{n-1}\}$-copy hypergraph of an n-vertex graph A, where P_{n-1} is a path of length $n-1$, has one hyperedge of size n if and only if there is a path in A that goes through all the vertices of A. This problem is known as the *Hamiltonian Path Problem* and is $\mathcal{NP}$-complete [42]. Also, the $\{K_m\}$-induced-copy hypergraph of an n-vertex graph A, where K_m is a complete graph with m vertices, has at least one hyperedge if and only if A has a clique of size m. This problem is known as the *Clique Problem* and is $\mathcal{NP}$-complete [42].

Therefore, when we are talking about finding an algorithm for a problem on a (an induced) copy hypergraph N, it is important that the input is the full representation of N, or A and $\mathcal{B}$ on which N is constructed. We call the latter representation the *base representation* of N. Moreover, we call $(A, \mathcal{B})$ a *base* of N.

We will look at the above-mentioned issue, together with the problem of finding intersperse colorings for copy hypergraphs, in Chapter 4.

2.3.4 Geometric Hypergraphs

Another family of hypergraphs that we are going to study in this thesis is one introduced by Smorodinsky [76]:

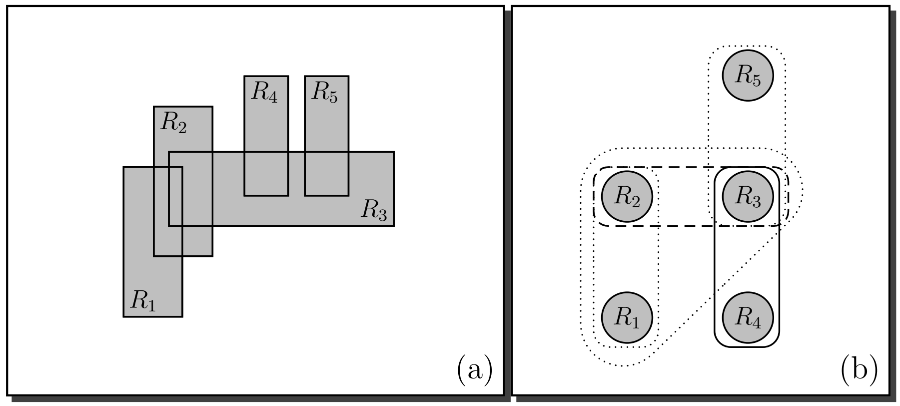

Figure 2.4: Part (a) shows a set of 2-dimensional boxes; part (b) illustrates the corresponding geometric hypergraph.

Definition 2.3.13. Suppose $\mathcal{R}$ is a set of k-dimensional regions, where by a *k-dimensional region* we mean a subset of $\mathbb{R}^k$. The *(geometric) hypergraph induced by* $\mathcal{R}$ is defined as the hypergraph $N_{\mathcal{R}} = (\mathcal{R}, \mathcal{E})$ where $\mathcal{E}$ consists of exactly those subsets $\mathcal{S}$ of $\mathcal{R}$ for which there exists a point that is in every region in $\mathcal{S}$ and no region in $\mathcal{R} - \mathcal{S}$. In other words, $\mathcal{E} = \left\{ \mathcal{S} \subseteq \mathcal{R} : \bigcap_{R \in \mathcal{S}} R - \bigcup_{R \notin \mathcal{S}} R \neq \emptyset \right\}$.

Note that, the geometric hypergraph reduced by $\mathcal{R}$ can be also obtained in the following way: we can draw the Venn diagram of $\mathcal{R}$ and take all subsets that are represented in the Venn diagram as hyperedges. Also, the geometric hypergraph induced by $\mathcal{R}$, $N_{\mathcal{R}} = (\mathcal{R}, \mathcal{E})$, is different from the more familiar concept "*intersection graph of* $\mathcal{R}$", which is a graph $A_{\mathcal{R}} = (\mathcal{R}, \mathcal{E}')$, where $\{R_1, R_2\} \in \mathcal{E}'$ if and only if $\{R_1, R_2\} \subseteq E$ for some $E \in \mathcal{E}$.

More restricted families of hypergraphs can be obtained by restricting the set of regions. A k-dimensional region B is called a *k-dimensional box* if there exist k ranges $[l_1, r_1]$, $[l_2, r_2]$, $\ldots$, $[l_k, r_k]$, where $l_i, r_i \in \mathbb{R}$, such that $B = \{(x_1, x_2, \ldots, x_k) : \forall 1 \leq i \leq d \quad x_i \in [l_i, r_i]\}$. Moreover, we call l_i (resp. r_i) the *left value* (resp. the *right value*) of the ith range of B. Figure 2.4 illustrates an example of a geometric hypergraph induced by 2-dimensional boxes. In this example, $\{R_1, R_2\}$ forms a hyperedge, because there is an area that is shared by only R_1 and R_2. Similarly, $\{R_1, R_2, R_3\}$ forms a hyperedge, because there is an area that is shared by only R_1, R_2, and R_3.

As will be explained in more detail in subsection 5.1, we reduce the problem of coloring the hypergraphs induced by axis-parallel boxes to a combinatorial problem based on permutations. The following definition deals with sets that can be constructed from permutations in a special way. Suppose we have a set of permutations $\{P_1, P_2, \ldots, P_k\}$ on V. Some subsets of V can be obtained by cutting off some elements from the head of these permutations. For example, if $P_1 = \langle 1, 3, 2, 4, 5\rangle$ and $P_2 = \langle 4, 2, 1, 5, 3\rangle$, we can obtain $\{2, 5\}$ by eliminating the first two elements of P_1 and the first element of P_2.

DEFINITION 2.3.14. Suppose V is a set and $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ is a set of k permutations on V. Then, $U \subseteq V$ is called *$\mathcal{P}$-constructible* if and only if there exist k cut-off points $a_1, a_2, \ldots, a_k$ $(0 \le a_i \le |V|)$ such that if we eliminate the first a_1 elements of P_1 from V, the first a_2 elements of P_2 from V, ..., and the first a_k elements of P_k from V, the set of remaining elements will be U.

As another example, if $V = \{v_1, v_2, \ldots, v_n\}$ and $\mathcal{P}$ consists of only one permutation $P_1 = \langle v_1, v_2, \ldots, v_n\rangle$, then the set of all $\mathcal{P}$-constructible sets is $\{V, \{v_2, \ldots, v_n\}, \{v_3, \ldots, v_n\}, \ldots, \{v_n\}, \emptyset\}$. If we add $P_2 = \langle v_n, v_{n-1}, \ldots, v_1\rangle$ to $\mathcal{P}$, then, in addition to the empty set, every set $\{v_i, v_{i+1}, \ldots, v_j\}$, $1 \le i \le j \le n$ will be $\mathcal{P}$-constructible because we can eliminate $\{v_1, v_2, \ldots, v_{i-1}\}$ using P_1 and then eliminate $\{v_n, v_{n-1}, \ldots, v_{j+1}\}$ using P_2. It is easy to see that these are all the possible $\mathcal{P}$-constructible sets.

A simple way to check if a set $U \subseteq V$ is $\mathcal{P}$-constructible is to search for an element o that is not in U but in each permutation one of the elements of U is before o. If there exists such an element, then U is not $\mathcal{P}$-constructible, since we have to eliminate at least one element from U to be able to eliminate o. Otherwise, U is $\mathcal{P}$-constructible, because for any element $o \notin U$ there is a permutation in which o is before all the elements of U, and hence, o can be eliminated using this permutation.

OBSERVATION 2.3.15. *$U \subseteq V$ is a $\mathcal{P}$-constructible set, where V is a set and $\mathcal{P} = \langle P_1, P_2, \ldots, P_k\rangle$ is a set of permutations on V, if and only if there is no element $o \in V - U$ such that in each permutation $P \in \mathcal{P}$ one of the elements of U is before o. We call such an o* an obstacle *to U.*

Proof. Recall that if U is $\mathcal{P}$-constructible, then there is a set of k cut-off points $a_1, a_2, \ldots, a_k$ such that if we eliminate the first a_i elements of P_i for each i, the set of remaining elements will be U. Therefore, for every $o \in V - U$ there exists an i such that o is in the first a_i elements of P_i. Also, no element of U is in the first a_i elements of P_i; otherwise, U would not be the set of remaining elements. Thus, there is no element $o \in V - U$ such that in each permutation $P \in \mathcal{P}$ one of the elements of U is before o.

On the other hand, if there is no element $o \in V - U$ such that in each permutation $P \in \mathcal{P}$ one of the elements of U is before o, then for every element $o \in V - U$, there are integers i_o and b_o such that o is in the b_oth position of P_{i_o} and all the elements of U are after the a_ith position. Therefore, if we set the k cut-off points as $a_i = \max\{b_o : o \in V - U \text{ and } i_o = i\}$ (if $i_o \neq i$ for all $o \in V - U$, we can set a_i to zero), the set of remaining elements will be U. Thus, U is $\mathcal{P}$-constructible.

□

In most cases, we only care about size-two $\mathcal{P}$-constructible sets. We can store size-two $\mathcal{P}$-constructible sets in a graph: the vertices of the graph are the elements in V and the edges are $\mathcal{P}$-constructible sets of size two. A more precise definition is the following.

DEFINITION 2.3.16. We call a graph A a *k-permutation-constructible graph (or a k-PC graph)* if there exists a set of k permutations $\mathcal{P}$ on $V(A)$ such that $\{u, v\} \in \mathcal{E}(A)$ if and only if $\{u, v\}$ is $\mathcal{P}$-constructible. We call G the *graph constructed on* $\mathcal{P}$.

In this thesis, if P is a permutation on V, $\mathcal{P}$ is a set of permutations on V, and $U \subseteq V$, we use $P[U]$ to denote the permutation on U induced by P, i.e. $x \in U$ is before $y \in U$ in $P[U]$ if and only if x is before y in P. We also use $\mathcal{P}[U]$ to denote $\{P[U] : P \in \mathcal{P}\}$.

2.4 Approaches

We will see in Chapter 3 that, unless $\mathcal{P} = \mathcal{NP}$, there is no polynomial-time algorithm for finding an ℓ-intersperse coloring for a given hypergraph that uses as few colors as possible. In most cases, this remains true if we restrict the input hypergraph to be from a specific family of hypergraphs. However, there exist other alternatives to attack an $\mathcal{NP}$-hard problem. One may find a fast algorithm, though exponential time, for an $\mathcal{NP}$-hard problem. We may also consider polynomial-time algorithms that can find an answer close enough to the correct answer of the problem, or that can find the correct answer with high probability. Another way to deal with these problems is to separate the "hard" part of the input instance and find an algorithm with fast running time ignoring the portion of the running time corresponding to the "hard" part of the input. We may change our complexity measure and try to find algorithms that have fast running times on the majority of input instances, according to some distribution among the input instances.

In this section, we introduce different approaches that we use.

2.4.1 Approximation Algorithms

In many situations, we face algorithmic problems in which the goal is to minimize or maximize a function, not just to decide whether the input is a "Yes" instance or a "No" instance. For example, the goal of the intersperse coloring problem is to find an ℓ-intersperse coloring for a given hypergraph that minimizes the number of colors. This kind of problem is called an *optimization problem* [81]. The following is a more precise definition.

Definition 2.4.1. A *maximization problem* (*minimization problem*) Π is a triple $(\mathcal{I}, \Lambda, f)$ in which $\mathcal{I}$ is the set of input instances, the function Λ maps every input instance to a set of possible solutions, and the function f maps every pair of (I, S), where $I \in \mathcal{I}$ is an input instance and $S \in \Lambda(I)$ is a solution to I, to a real number. The goal of a maximization problem (minimization problem) is to find a solution S_I for a given input instance $I \in \mathcal{I}$ such that $f(I, S_I)$ is maximized (minimized) over all pairs (I, S), for all $S \in \Lambda(I)$. The value of $f(I, S_I)$ is denoted by $OPT_\Pi(I)$.

An *optimization problem* is either a maximization problem or a minimization problem.

When finding an algorithm for computing the exact optimal solution of an optimization problem is hard, it is reasonable to seek an algorithm that returns an "almost" optimal solution.

Definition 2.4.2. An *approximation algorithm* for an optimization problem $\Pi = (\mathcal{I}, \Lambda, f)$ is a deterministic polynomial-time algorithm A that for every input $I \in \mathcal{I}$ $A(I) \in \Lambda(I)$. The *approximation ratio* of A is

$$\alpha(n) = \begin{cases} \max_{I \in \mathcal{I}, |I| = n} \frac{OPT_\Pi(I)}{f(I, A(I))} & \text{if } \Pi \text{ is a maximization problem} \\ \\ \max_{I \in \mathcal{I}, |I| = n} \frac{f(I, A(I))}{OPT_\Pi(I)} & \text{if } \Pi \text{ is a minimization problem} \end{cases} .$$

A is a $\beta(n)$-*approximation* for Π if the approximation ratio of A is at most $\beta(n)$, for all integers n.

Finding approximation algorithms for proper coloring of hypergraphs has been extensively studied [8, 25, 46, 52, 54, 59]. Alon et al. developed an approximation algorithm with ratio $O(n^{1-\frac{1}{r}} \log^{1-\frac{1}{r}} n)$ for proper coloring of 2-colorable hypergraphs, where r is the rank of the hypergraph [8]. Note that even checking if a given hypergraph is 2-colorable is $\mathcal{NP}$-complete [63].

As for uniform hypergraphs, the latest result is due to Krivelevich and Sudakov [59] who developed an approximation algorithm with ratio $O(\frac{n(\log\log n)^2}{\log^2 n})$ for coloring uniform hypergraphs. Their approach is based on Alon et al.'s generalized degree reduction technique.

There are also hardness results for approximation algorithms. Krivelevich and Sudakov extended the well-known inapproximability result on graph coloring [48] and proved the chromatic number of uniform hypergraphs cannot be approximated within ratio $O(n^{1-\varepsilon})$, for any $\varepsilon > 0$, unless $\mathcal{NP} \subseteq \mathcal{ZPP}$ [59], where

$\mathcal{ZPP} = \mathcal{RP} \cap \text{co}\mathcal{RP}$,

$\mathcal{RP}$ is the class of languages L that can be decided by a polynomial-time probabilistic algorithm A such that A accepts any $x \in L$ with probability at least $\frac{1}{2}$ and rejects any $x \notin L$ with probability 1, and

co$\mathcal{RP}$ is the class of languages L such that $\overline{L} \in \mathcal{RP}$.

Also Khot proved that there exists a constant c such that no polynomial-time algorithm can color a p-colorable 4-uniform hypergraph with $O(\log^{cp} n)$ colors, for any integer $p \geq 7$ [54].

In this thesis, we examine approximation algorithms for the intersperse coloring problem in different hypergraph classes.

2.4.2 Property Testing Algorithms

It is interesting that a number of complicated combinatorial problems can be solved efficiently when we allow a small probability of error and we require an approximately correct answer instead of a strictly correct answer. Among the computation models motivated by the above fact is the property testing model introduced by Rubinfeld and Sudan [72]. The following is the intuitive definition of property testing suggested by Goldreich et al. [43]:

> *Property testing* is the study of the following class of problems: given oracle access to a function f, we want to probabilistically find out whether f has a specific property P or it is far from any function that has property P.

This definition is not precise. For example, no method for measuring the distance between two objects is mentioned. We give a more precise definition later in this section.

Testing graph properties is a special case of property testing in which the function f "represents" a graph and the property P is a graph property [43]; a *(hyper)graph property* is simply a set of (hyper)graphs. There are four models for testing graph properties: the adjacency matrix model [43], the bounded degree incidence list model [44], the unbounded degree incidence list model [70], and the mixed model [53]. These models differ in the way they define how f "represents" a graph and how the distance between two graphs is computed. The first model is suitable for dense graphs, the second and third models are suitable for non-dense graphs, and the fourth model, which is a combination of the first and the third models, is suitable for both dense and non-dense graphs. However, all these models are non-symmetric in the sense that they allow the tester algorithm to "probe" only vertices, not edges. In fact, some hardness results in the area of graph property testing are based on this non-symmetry of models.

In this thesis, we generalize the mixed model of Kaufman et al. [53] in a natural way and propose a symmetric model, suitable for both dense and non-dense graphs, that allows the tester algorithm to probe vertices and edges. We show that the symmetric model is more powerful than the mixed model by proving that there exists a graph property P such that P has a tester algorithm in the symmetric model that runs in time $O(\text{poly}(1/\varepsilon))$, but there is a lower bound of $\Omega(n^{1/2})$ for the running time of any tester algorithm for P in the mixed model; where n is the number of vertices of the input graph, and ε is the error bound and is given as a part of the input. For exact definitions of the terms, refer to Definition 2.4.5.

Another contribution of the symmetric model is the function it uses for computing the distance between two graphs. The advantage of this distance function is that it can be applied to non-uniform hypergraphs. Moreover, for graphs and uniform hypergraphs, it is equivalent to the distance function used in the unbounded degree incidence list model [70] and in the mixed model [53].

As mentioned above, one of the main differences among different property testing models is the way they define the distance function between two graphs. In this work, our distance function is a generalization of the one proposed by Parnas and Ron [70]. They proposed the following distance function between two graphs $N = (V_1, \mathcal{E}_1)$ and $O = (V_2, \mathcal{E}_2)$:

$$PRDist(N, O) = \frac{\text{minimum number of edge insertions/deletions needed to convert } N \text{ to } O}{|\mathcal{E}_1|}$$

Note that *PRDist*(N,O) is not necessarily equal to *PRDist*(O,N), and *PRDist*(N,O) can be greater than one. Also, to avoid redundant technicalities, we assume that $V_1 = V_2$; for graphs with different vertex sets, we define *PRDist*$(N,O) = \infty$. Although the above definition of *PRDist* works well for graphs, it is not suitable for non-uniform hypergraphs, because, in the above definition, inserting a size-two hyperedge has the same cost as inserting an edge of size n. The following formula is better because it assigns different weights to different insertions and deletions.

$$\frac{\sum\limits_{E \in \mathcal{E}_1 \triangle \mathcal{E}_2} |E|}{\sum\limits_{E \in \mathcal{E}_1} |E|} \tag{2.1}$$

where $\mathcal{E}_1 \triangle \mathcal{E}_2$ is the *symmetric difference* between $\mathcal{E}_1$ and $\mathcal{E}_2$, defined in the following way:

$$\mathcal{E}_1 \triangle \mathcal{E}_2 = (\mathcal{E}_1 - \mathcal{E}_2) \cup (\mathcal{E}_2 - \mathcal{E}_1).$$

Thus, $\mathcal{E}_1 \triangle \mathcal{E}_2$ is exactly the set of edges needed to be inserted or deleted to convert $\mathcal{E}_1$ to $\mathcal{E}_2$. Note that, due to the definition of *PRDist*(N,O), *PRDist*(N,O) is the minimum number of edge insertions and deletions needed to convert N to O, divided by the number of edges of N. Therefore,

$$\mathit{PRDist}(N,O) = \frac{|(\mathcal{E}(N) - \mathcal{E}(O)) \cup (\mathcal{E}(O) - \mathcal{E}(N))|}{|\mathcal{E}(N)|} = \frac{|\mathcal{E}(N) \triangle \mathcal{E}(O)|}{|\mathcal{E}(N)|}.$$

Distance function 2.1 still has another drawback: it only considers insertions and deletions of edges. A more general approach could allow inserting vertices into edges and deleting vertices from edges, too. Therefore, we take $|E_1 \triangle E_2|$ as the cost of converting a hyperedge E_1 to another hyperedge E_2. To formalize the cost of converting a set of hyperedges $\mathcal{E}_1$ to another set of hyperedges $\mathcal{E}_2$, we consider all possible transformations from $\mathcal{E}_1$ to $\mathcal{E}_2$; a *transformation* from $\mathcal{E}_1$ to $\mathcal{E}_2$ is a relation $T \subseteq \{\mathcal{E}_1 \cup \{\emptyset\}\} \times \{\mathcal{E}_2 \cup \{\emptyset\}\}$ such that for any $E \in \mathcal{E}_1$ there is exactly one pair $(E, X) \in T$ and for any $E \in \mathcal{E}_2$ there is exactly one pair $(X, E) \in T$. Thus, $(E_1, E_2) \in T$ means that the hyperedge E_1 should be converted to the hyperedge E_2. Also, $(E_1, \emptyset) \in T$ (or $(\emptyset, E_2) \in T$) means that E_1 should be completely deleted (or E_2 should be completely inserted). Then, the cost of converting $\mathcal{E}_1$ to $\mathcal{E}_2$, denoted by $\psi(\mathcal{E}_1, \mathcal{E}_2)$, is defined to be the minimum of $\sum_{(E_1,E_2) \in T} |E_1 \triangle E_2|$ over all transformations T from $\mathcal{E}_1$ to $\mathcal{E}_2$. Now, we define our distance function between two hypergraphs N and O as

$$\mathit{SymDist}(N,O) = \frac{\psi(\mathcal{E}(N), \mathcal{E}(O))}{\sum\limits_{E \in \mathcal{E}(N)} |E|}.$$

For example, if $\mathcal{E}(N) = \{\{v_1, v_2, v_3\}, \{v_2, v_4\}\}$ and $\mathcal{E}(O) = \{\{v_1, v_2, v_4\}, \{v_1, v_5\}\}$, then $\mathit{SymDist}(N, O) = \frac{4}{5}$, because the optimum transformation is $\{(\{v_1, v_2, v_3\}, \{v_1, v_5\}), (\{v_2, v_4\}, \{v_1, v_2, v_4\})\}$. Note that *SymDist* is not a symmetric function. The name stands for the distance function of the symmetric model.

We generalize the notation of distance functions, *PRDist* and *SymDist*, to measure the distance between a hypergraph and a property:

Definition 2.4.3. If N is a hypergraph and P is a property, then we use $\mathit{SymDist}(N, P)$ (resp. $\mathit{PRDist}(N, P)$) to denote $\min_{O \in P} \mathit{SymDist}(N, O)$ (resp. $\min_{O \in P} \mathit{PRDist}(N, O)$).

The following lemma shows that our distance function behaves similarly to *PRDist* on graphs.

Lemma 2.4.4. *If $N = (V, \mathcal{E}_N)$ and $O = (V, \mathcal{E}_O)$ are two 2-uniform hypergraphs, then $\mathit{PRDist}(N, O)/2 \leq \mathit{SymDist}(N, O) \leq \mathit{PRDist}(N, O)$.*

Proof. Recall that

$$\mathit{PRDist}(N, O) = \frac{|\mathcal{E}_N \triangle \mathcal{E}_O|}{|\mathcal{E}_N|}.$$

Consider the transformation T from $\mathcal{E}_N$ to $\mathcal{E}_O$ that includes (E_1, E_2) if $E_1 = E_2$, $(E_1, \emptyset)$ if $E_1 \in \mathcal{E}_N - \mathcal{E}_O$, and $(\emptyset, E_2)$ if $E_2 \in \mathcal{E}_O - \mathcal{E}_N$. Clearly, $\psi(\mathcal{E}_N, \mathcal{E}_O)$ is at most

$$\sum_{(E_1,E_2) \in T} |E_1 \triangle E_2| = \sum_{E_1 \in \mathcal{E}_N - \mathcal{E}_O} 2 + \sum_{E_2 \in \mathcal{E}_O - \mathcal{E}_N} 2 = 2 \cdot |\mathcal{E}_N \triangle \mathcal{E}_O|.$$

Hence,

$$\mathit{SymDist}(N, O) \leq \frac{2 \cdot |\mathcal{E}_N \triangle \mathcal{E}_O|}{\sum_{E \in \mathcal{E}_N} 2} = \frac{|\mathcal{E}_N \triangle \mathcal{E}_O|}{|\mathcal{E}_N|} = \mathit{PRDist}(N, O).$$

On the other hand, suppose that T^* is the transformation from $\mathcal{E}_N$ to $\mathcal{E}_O$ that minimizes $\sum_{(E_1,E_2) \in T} |E_1 \triangle E_2|$, i.e. $\psi(\mathcal{E}_N, \mathcal{E}_O) = \sum_{(E_1,E_2) \in T^*} |E_1 \triangle E_2|$. From the definition of transformation, we know that T^* has at least $(|\mathcal{E}_N| + |\mathcal{E}_O|)/2$ pairs. Suppose that T^* has exactly k pairs (E_1, E_2) such that $E_1 = E_2$. Then, T^* has at least $(|\mathcal{E}_N| + |\mathcal{E}_O|)/2 - k$ pairs (E_1, E_2) such that $E_1 \neq E_2$. Because each one of E_1 and E_2 is either a size-two set or the empty set, if $E_1 \neq E_2$ then $|E_1 \triangle E_2| \geq 2$.

Therefore, considering the fact that $k \leq |\mathcal{E}_N \cap \mathcal{E}_O|$, we have

$$\begin{aligned}
\psi\left(\mathcal{E}_N, \mathcal{E}_O\right) &\geq \left(\frac{|\mathcal{E}_N| + |\mathcal{E}_O|}{2} - k\right) \times 2 \\
&= |\mathcal{E}_N \cup \mathcal{E}_O| + |\mathcal{E}_N \cap \mathcal{E}_O| - 2k \\
&\geq |\mathcal{E}_N \cup \mathcal{E}_O| - |\mathcal{E}_N \cap \mathcal{E}_O| \\
&= |\mathcal{E}_N \triangle \mathcal{E}_O|.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathit{SymDist}(N, O) &= \frac{\psi(\mathcal{E}_N, \mathcal{E}_O)}{\sum_{E \in \mathcal{E}_N} |E|} \\
&\geq \frac{|\mathcal{E}_N \triangle \mathcal{E}_O|}{2|\mathcal{E}_N|} \\
&= \frac{\mathit{PRDist}(N, O)}{2}.
\end{aligned}$$

□

We say that a hypergraph N is *ε-far* from a property P if $\mathit{SymDist}(N, P) > \varepsilon$; otherwise, we call N *ε-close* to P.

DEFINITION 2.4.5. Suppose P is a hypergraph property and $\mathcal{C}$ is a class of hypergraphs. Then, a *hypergraph property tester* for P on $\mathcal{C}$ *in the symmetric model* is a probabilistic algorithm A such that for any hypergraph $N = (V, \mathcal{E}) \in \mathcal{C}$ and any ordering of V, on input parameters $|V|$, $|\mathcal{E}|$, and $0 < \varepsilon \leq 1$

1. If $N \in P$, A accepts with probability at least 2/3.
2. If N is ε-far from P, A accepts with probability at most 1/3.

Moreover, A is allowed to submit queries of the following forms, for $U \subseteq V$, $v \in V$, and $E \in \mathcal{E}$:

1. "Does $\mathcal{E}$ contain U?".
2. "How many edges contain v?".
3. "How many vertices does E have?".
4. "What is the ith edge that contains v?".

5. "What is the ith vertex of E?".

The number of queries submitted by the algorithm is called the *query complexity* of the algorithm.

Note that in the above definition, the number 1/3 (2/3) does not have any special property; it can be replaced by any constant less (more) than 1/2. Therefore, in our model, which will be referred to as the *symmetric model*, a hypergraph property tester can submit queries about edges, i.e. query types 3 and 5, called *edge queries*, as well as queries about vertices.

The followings are the definitions of existing property testing models. In all the following definitions, P is a graph property and $\mathcal{C}$ is a class of graphs.

DEFINITION 2.4.6. ([43]) A *graph property tester* for P on $\mathcal{C}$ *in the adjacency matrix model* is a probabilistic algorithm A such that for any graph $A = (V, \mathcal{E}) \in \mathcal{C}$, on input parameters $|V|$ and $0 < \varepsilon \leq 1$

1. If $A \in P$, A accepts with probability at least 2/3.
2. If at least εn^2 edge modifications are need to convert A to a graph in P, A accepts with probability at most 1/3.

Moreover, A is allowed to submit queries of the form "Is $\{u, v\} \in \mathcal{E}$?", for $u, v \in V$.

DEFINITION 2.4.7. ([44]) A *graph property tester* for P on $\mathcal{C}$ *in the bounded degree incidence list model* is a probabilistic algorithm A such that for any graph $A = (V, \mathcal{E}) \in \mathcal{C}$ with maximum degree d and any ordering of V, on input parameters $|V|$, d, and $0 < \varepsilon \leq 1$

1. If $A \in P$, A accepts with probability at least 2/3.
2. If at least εdn edge modifications are need to convert A to a graph in P, A accepts with probability at most 1/3.

Moreover, A is allowed to submit queries of the form "What is the ith neighbor of u?", for $u \in V$.

DEFINITION 2.4.8. ([70]) A *graph property tester* for P on $\mathcal{C}$ *in the unbounded-degree incidence list model* is a probabilistic algorithm A such that for any graph $A = (V, \mathcal{E}) \in \mathcal{C}$ and any ordering of V, on input parameters $|V|$ and $0 < \varepsilon \leq 1$

1. If $A \in P$, A accepts with probability at least $2/3$.

2. If at least $\varepsilon|\mathcal{E}|$ edge modifications are need to convert A to a graph in P, A accepts with probability at most $1/3$.

Moreover, A is allowed to submit queries of the following forms, for $v \in V$:

1. "How many edges contain v?".

2. "What is the ith neighbor of v?".

DEFINITION 2.4.9. ([53]) A *graph property tester* for P on $\mathcal{C}$ *in the mixed model* is a probabilistic algorithm A such that for any graph $A = (V, \mathcal{E}) \in \mathcal{C}$ and any ordering of V, on input parameters $|V|$ and $0 < \varepsilon \leq 1$

1. If $A \in P$, A accepts with probability at least $2/3$.

2. If at least $\varepsilon|\mathcal{E}|$ edge modifications are need to convert A to a graph in P, A accepts with probability at most $1/3$.

Moreover, A is allowed to submit queries of the following forms, for $u, v \in V$:

1. "How many edges contain v?".

2. "What is the ith neighbor of v?".

3. "Is u connected to v?".

To give better intuition about the symmetric model, we use the *bipartite incidence structure* of hypergraphs [75], which is analogous to the incidence list of classical graphs: the bipartite incidence structure of a hypergraph $N = (V, \mathcal{E})$ consists of $|V| + |\mathcal{E}|$ lists such that each vertex and hyperedge points to a distinct list. Additionally,

1. The list corresponding to a vertex $v \in V$ has $d_N(v)$ members. The ith member is a pointer to the list corresponding to the ith hyperedge that contains v.

2. The list corresponding to a hyperedge $E \in \mathcal{E}$ has $|E|$ members. The ith member is a pointer to the list corresponding to the ith vertex of E.

Therefore, in the symmetric model, we allow the tester to submit queries about the adjacency matrix (query type 1) and the bipartite incidence structure of the input hypergraph (query types 2 to 5). Note that in the bipartite incidence structure, we assumed that an arbitrary order among the vertices and the hyperedges exists. Thus, the symmetric model is a natural generalization of the mixed model in which the tester is allowed to submit queries about the adjacency matrix and the incidence list of the input graph.

Chapter 3

General Hypergraphs

In this chapter we obtain results for hypergraphs in general. We obtain two types of results: upper bounds and approximation algorithms. In Section 3.1, we find a number of general upper bounds for the ℓ-intersperse chromatic number of hypergraphs, for any integer $\ell \geq 2$. Then, in Section 3.2, we develop a simple approximation algorithm that colors any q-colorable hypergraph with $O(\frac{n}{\lg n})$ colors in polynomial time, for any constant q.

3.1 Upper Bounds

The most general results of this thesis, about the ℓ-intersperse coloring of hypergraphs, are presented in this section. We put no limitation on hypergraphs or the value of ℓ. Naturally, the results are not so strong in the sense that it is possible to find better results if we put restrictions on hypergraphs or the value of ℓ.

The next theorem uses a probabilistic argument to prove a general upper bound on the ℓ-intersperse chromatic number of hypergraphs. Later, in Theorem 3.1.5, we will derandomize Theorem 3.1.2 and present a constructive version. We need the following notation in Theorem 3.1.2 and Theorem 3.1.5.

Definition 3.1.1. Suppose $0 \leq \alpha, \delta \leq 1$ are real numbers and n is an integer. Then, $p_{n,\delta}^{(\alpha)}$ denotes the probability that we observe more than $\delta n - \delta$ heads when we toss a biased coin n times, where the probability of tossing a head is α in each single step.

Theorem 3.1.2. *Suppose $N = (V, \mathcal{E})$ is a hypergraph and $\ell \geq 2$ is an integer.*

Then, for every integer $c \geq 1$,

$$\chi_\ell(N) \leq c + c \sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}},$$

where $p^{(\alpha)}_{n,\delta}$ is defined in Definition 3.1.1.

Proof. We use a probabilistic proof. For each vertex of N, we choose a random color uniformly from $\{1, 2, \ldots, c\}$. At this phase, some hyperedges may not be ℓ-full colored. We call such hyperedges *bad hyperedges*. We will recolor some vertices such that at the end there will be no bad hyperedge. Let $\beta_E = \frac{1}{\min\{|E|,\ell\}-1}$. Instead of focusing on bad hyperedges, we consider any hyperedge $E \in \mathcal{E}$ that has more than $\beta_E(|E| - 1)$ vertices with the same color b for some $1 \leq b \leq c$. Note that if a hyperedge $E \in \mathcal{E}$ has at most $\beta_E(|E| - 1)$ vertices with the same color, then E has at least $\min\{|E|, \ell\}$ different colors, because the number of different colors is at least

$$\begin{aligned}
\frac{|E|}{\beta_E(|E|-1)} &= \frac{(\min\{|E|,\ell\}-1)|E|}{|E|-1} \\
&= \frac{(|E|-1)\min\{|E|,\ell\} - (|E| - \min\{|E|,\ell\})}{|E|-1} \\
&= \min\{|E|,\ell\} - \frac{|E| - \min\{|E|,\ell\}}{|E|-1} \\
&> \min\{|E|,\ell\} - 1.
\end{aligned}$$

Since the number of different colors of E is an integer, it is at least $\min\{|E|, \ell\}$. Thus, all bad hyperedges have more than $\beta_E(|E| - 1)$ vertices with the same color. Consequently, it is enough to make sure that any hyperedge $E \in \mathcal{E}$ that has more than $\beta_E(|E| - 1)$ vertices with the same color gets at least $\min\{|E|, \ell\}$ different colors.

In the next phase, we fix the coloring of the hyperedges that have more than $\beta_E(|E| - 1)$ vertices of the same color. For a hyperedge $E \in \mathcal{E}$ and a color $b \in \{1, 2, \ldots, c\}$, let $C_{E,b}$ be the set of vertices of E with color b. If $C_{E,b}$ has more than $\beta_E(|E| - 1)$ vertices, we partition $C_{E,b}$ into $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil$ disjoint subsets of sizes at most $\beta_E(|E|-1)$ and assign a different unused color to each of the first $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil -$ 1 subsets. Clearly, after doing the above-mentioned recoloring procedure for all hyperedges and all colors, there will be no bad hyperedge.

To estimate the number of new colors we use the following method. For a hyperedge $E \in \mathcal{E}$, a color $1 \leq b \leq c$, and a real number $0 \leq \delta \leq 1$, we set the random variable $X_{E,b,\delta}$ to one if E has more than $\delta(|E|-1)$ vertices of color b; otherwise, we set $X_{E,b,\delta}$ to zero. The number of new colors is exactly the sum of $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1$ over all hyperedges $E \in \mathcal{E}$ and colors $1 \leq b \leq c$ such that $C_{E,b}$ has more than $\beta_E(|E|-1)$ vertices. However, if $|C_{E,b}| > \beta_E(|E|-1)$, then $X_{E,b,i\beta_E} = 1$ for all $1 \leq i \leq \left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1$ and $X_{E,b,i\beta_E} = 0$ for all $i > \left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1$. Hence, we have $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1 = \sum_{i=1}^{\min\{|E|,b\}-1} X_{E,b,i\beta_E}$. Therefore,

$$\begin{aligned} \text{Number of new colors} &= \sum_{E\in\mathcal{E}} \sum_{b \text{ s.t. } |C_{E,b}|>\beta_E(|E|-1)} \left(\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1 \right) \\ &= \sum_{E\in\mathcal{E}} \sum_{b=1}^{c} \sum_{i=1}^{\min\{|E|,\ell\}-1} X_{E,b,i\beta_E}. \end{aligned}$$

Since $\Pr[X_{E,b_1,\delta} = 1] = \Pr[X_{E,b_2,\delta} = 1]$ for all $1 \leq b_1, b_2 \leq c$, we can simplify the above formula to $c \sum_{E\in\mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} X_{E,1,i\beta_E}$. So, the expected number of new colors is

$$c \sum_{E\in\mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} \Pr[X_{E,1,i\beta_E} = 1] \quad = \quad c \sum_{E\in\mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}} \qquad (3.1)$$

because $X_{E,1,i\beta_E}$ is one if and only if more than $\frac{i}{\min\{|E|,\ell\}-1}(|E|-1)$ vertices of E are colored with 1, and we know that each vertex is colored with 1 with probability $\frac{1}{c}$, by Definition 3.1.1. Thus, $X_{E,1,i\beta_E}$ is one with probability $p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}}$. As 3.1 is the expected number of new colors, there exists a coloring in the first phase for which the number of new colors in the second phase does not exceed $c \sum_{E\in\mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}}$. Hence, the total number of colors will be at most $c + c \sum_{E\in\mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}}$. □

The following corollary will help us understand the impact of Theorem 3.1.2 better.

COROLLARY 3.1.3. *Suppose A is a graph. Then,* $\chi(A) \leq \left\lceil 2\sqrt{|\mathcal{E}(A)|} \right\rceil$.

Proof. Since a graph is a 2-uniform hypergraph, we can use Theorem 3.1.2 and set $|E| = 2$ for all hyperedges and $\ell = 2$. Then, for all $c \geq 1$ we have

$$\chi(A) \leq c + c \sum_{E \in \mathcal{E}(A)} p_{2,1}^{(\frac{1}{c})}.$$

Therefore, $p_{2,1}^{(\frac{1}{c})} = \frac{1}{c^2}$ and so $\chi(A) \leq c + \frac{|\mathcal{E}(A)|}{c}$. We set $c = \left\lceil \sqrt{|\mathcal{E}(A)|} \right\rceil$ and the corollary follows immediately:

$$\chi(A) \leq \left\lceil \sqrt{|\mathcal{E}(A)|} \right\rceil + \frac{|\mathcal{E}(A)|}{\left\lceil \sqrt{|\mathcal{E}(A)|} \right\rceil} \leq \left\lceil 2\sqrt{|\mathcal{E}(A)|} \right\rceil$$

□

Corollary 3.1.3 does not give a new result. For example, consider the following simpler argument: for a proper coloring of A with $\chi(A)$ colors and any two colors $1 \leq i < j \leq \chi(A)$ there is an edge in A whose ends are colored with i and j; otherwise, we could color A with $\chi(A) - 1$ colors by merging the color class i with the color class j. Therefore, A has at least $\frac{1}{2}\chi(A)(\chi(A) - 1)$ edges, which means $\chi(A) \leq \frac{1}{2} + \sqrt{\frac{1}{4} + 2|\mathcal{E}(A)|}$. On the other hand, this upper bound cannot be improved substantially in terms of the number of edges, because the chromatic number of a graph A can be as large as $\frac{1}{2} + \sqrt{\frac{1}{4} + 2|\mathcal{E}(A)|}$ if A is a complete graph.

The following corollary is more general than Corollary 3.1.3. It gives an upper bound on the chromatic number of uniform hypergraphs based on the number of hyperedges.

Corollary 3.1.4. *Suppose N is an r-uniform hypergraph. Then,*

$$\chi(N) \quad \leq \quad \left\lceil 2|\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil.$$

Proof. We set $\ell = 2$ and $|E| = r$ for all hyperedges in Theorem 3.1.2. Thus, we have

$$\begin{aligned} \chi(N) \quad &\leq \quad c + c \sum_{E \in \mathcal{E}(N)} p_{r,1}^{(\frac{1}{c})} \\ &= \quad c + c|\mathcal{E}(N)| p_{r,1}^{(\frac{1}{c})}. \end{aligned}$$

Thus, $p_{r,1}^{(\frac{1}{c})} = \frac{1}{c^r}$ and $\chi(N) \leq c + \frac{|\mathcal{E}(N)|}{c^{r-1}}$. We set $c = \left\lceil |\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil$ and the corollary follows immediately:

$$\chi(N) \leq \left\lceil |\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil + \frac{|\mathcal{E}(N)|}{\left\lceil |\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil^{r-1}} \leq \left\lceil 2|\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil$$

□

Again, Corollary 3.1.4 does not give a new result. Kostochka et al. proved a stronger theorem [57]: they showed that the chromatic number of an r-uniform hypergraph N is at most

$$\gamma \cdot \frac{|\mathcal{E}(N)|^{\frac{1}{r}}}{(\ln|\mathcal{E}(N)|)^{\frac{1}{r-1}}},$$

where γ is a constant. The also proved that their bound is tight up to a constant factor. However, they used a complicated probabilistic method and did not mention whether their method can be turned into a deterministic polynomial-time algorithm for finding a proper coloring with at most $c \cdot \frac{|\mathcal{E}(N)|^{\frac{1}{r}}}{(\ln|\mathcal{E}(N)|)^{\frac{1}{r-1}}}$ colors.

For the strong coloring problem the upper bound obtained by Agnarsson and Halldórsson is $r\sqrt{|\mathcal{E}(N)|}$, where r is the rank of N [3]. However, Theorem 3.1.2 gives a better bound if N has few hyperedges with size close to r. For example, suppose N has x hyperedges of size r and all the other hyperedges are of size two. A strong coloring of N is equivalent to an intersperse coloring of N if we set $\ell = r$. Then, using Theorem 3.1.2 we can say that N can be colored with at most

$$c + c \sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,r\}-1} p_{|E|,\frac{i}{\min\{|E|,r\}-1}}^{(\frac{1}{c})}$$

colors, where c is any integer. Here, $\sum_{i=1}^{\min\{|E|,r\}-1} p_{|E|,\frac{i}{\min\{|E|,r\}-1}}^{(\frac{1}{c})}$ is $\frac{1}{c^2}$ if $|E| = 2$ and is

$$\begin{aligned} \sum_{i=2}^{r} (i-1) \binom{r}{i} \left(\frac{1}{c}\right)^i \left(1 - \frac{1}{c}\right)^{r-i} &\leq (r-1) \sum_{i=2}^{r} \binom{r}{i} \left(\frac{1}{c}\right)^i \left(1 - \frac{1}{c}\right)^{r-i} \\ &\leq r - 1 \end{aligned}$$

if $|E| = r$. So, if we set $c = \sqrt{\frac{|\mathcal{E}(N)|-x}{xr-x+1}}$, we get the upper bound of

$$\frac{|\mathcal{E}(N)| - x}{\sqrt{\frac{|\mathcal{E}(N)|-x}{xr-x+1}}} + \sqrt{\frac{|\mathcal{E}(N)| - x}{xr - x + 1}} \cdot (xr - x + 1) \leq 2\sqrt{(|\mathcal{E}(N)| - x)(xr - x + 1)},$$

which is better than $r\sqrt{|\mathcal{E}(N)|}$ if $x \leq \frac{r}{4}$.

Below, we prove that Theorem 3.1.2 can be derandomized.

THEOREM 3.1.5. *There exists a polynomial-time algorithm that, for a given hypergraph $N = (V, \mathcal{E})$ and given integers $\ell \geq 1$ and $c \geq 1$, computes an ℓ-intersperse coloring with at most*

$$c + c \sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|, \frac{i}{\min\{|E|,\ell\}-1}}$$

colors.

Proof. We use the method of conditional expectations [31, 11], which is a general method for derandomization, to derandomize the method explained in the proof of Theorem 3.1.2.

The main idea behind the method of conditional expectations is the following: suppose that after a sequence of random choices the expected value of a goal function is **E**. Then, in many cases, it is possible to replace the random choices by polynomial-time deterministic choices such that the expected value of the goal function after the ith choice, assuming that the remaining choices will be made randomly, does not exceed **E**. We show that this method can be successfully applied to Theorem 3.1.2.

Our deterministic algorithm has two phases, analogous to the phases of the randomized algorithm described in the proof of Theorem 3.1.2. We only derandomize the first phase, as the second phase is already deterministic. For the first phase, instead of randomly selecting the color of each vertex, we fix an arbitrary order, and color the vertices in a deterministic way so that the number of new colors in the second phase is at most

$$c \textstyle\sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|, \frac{i}{\min\{|E|,\ell\}-1}}. \tag{3.2}$$

Suppose $\langle v_1, v_2, \ldots, v_n \rangle$ is an arbitrary order on V, where $n = |V|$. We set the color of v_1 to $\ell_1 = 1$. Suppose we have colored $v_1, v_2, \ldots, v_t$ with colors $\ell_1, \ell_2, \ldots, \ell_t$, other vertices are not colored, and we are about to color v_{t+1}. For every hyperedge $E \in \mathcal{E}$ and color $1 \leq b \leq c$, we use $f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t)$ to denote the number of vertices of E that are colored with b given that the colors of $v_1, v_2, \ldots, v_t$ are $\ell_1, \ell_2, \ldots, \ell_t$ and other vertices are not colored. Again, let $X_{E,b,\delta}$ be the event that the hyperedge $E \in \mathcal{E}$ has more than $\delta(|E| - 1)$ vertices with color b, and

$\Pr[X_{E,b,\delta} = 1 \mid \ell_1, \ell_2, \ldots, \ell_t]$ be the probability that $X_{E,b,\delta} = 1$ when $v_1, v_2, \ldots, v_t$ are colored with $\ell_1, \ell_2, \ldots, \ell_t$ and the color of other vertices are chosen independently and uniformly from $\{1, 2, \ldots, c\}$. Then,

$$\Pr[X_{E,b,\delta} = 1 \mid \ell_1, \ldots, \ell_t] = \begin{cases} 1 & \text{if } \delta(|E|-1) - f(E, b \mid \ell_1, \ldots, \ell_t) < 0 \\ 0 & \text{if } \delta(|E|-1) - f(E, b \mid \ell_1, \ldots, \ell_t) \geq |E - \{v_1, \ldots, v_t\}| \\ \frac{1}{c} & \text{if } |E - \{v_1, \ldots, v_t\}| = 1 \text{ and } \delta(|E|-1) - f(E, b \mid \ell_1, \ldots, \ell_t) = 0, \text{ and} \\ p^{(\frac{1}{c})}_{|E-\{v_1,\ldots,v_t\}|,\delta'} & \text{otherwise} \end{cases}$$

where $\delta' = \frac{\delta(|E|-1) - f(E,b|\ell_1,\ell_2,\ldots,\ell_t)}{|E-\{v_1,v_2,\ldots,v_t\}|-1}$. There are four cases:

1. In the first case of the above equation, where $\delta(|E|-1) - f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t) < 0$, the hyperedge E has more than $\delta(|E|-1)$ vertices of color b among the first t vertices. So, no matter how the colors of the other vertices are chosen $X_{E,b,\delta}$ is always one.

2. In the second case, where $\delta(|E|-1) - f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t) \geq |E - \{v_1, v_2, \ldots, v_t\}|$ only $f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t)$ vertices of E, that are among the first t vertices, are colored with b and even if all the remaining vertices are colored with b, $X_{E,b,\delta}$ will be zero.

3. In the third case, where E has only one uncolored vertex and it has $\delta(|E|-1)$ vertices with color b, $X_{E,b,\delta}$ will be one if and only if the only uncolored vertex will be assigned to b. Thus the probability that $X_{E,b,\delta} = 1$ is $\frac{1}{c}$.

4. Finally, where $0 \leq \delta(|E|-1) - f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t) < |E - \{v_1, v_2, \ldots, v_t\}|$ and E has more than one uncolored vertex, $X_{E,b,\delta}$ is one if and only if more than $\delta(|E|-1) - f(E, b \mid \ell_1, \ell_2, \ldots, \ell_t) = \delta'(|E - \{v_1, v_2, \ldots, v_t\}| - 1)$ uncolored vertices of E are colored with b in the process of randomly coloring $v_{t+1}, v_{t+2}, \ldots, v_n$. Therefore, the conditional probability is $p^{(\frac{1}{c})}_{|E-\{v_1,v_2,\ldots,v_t\}|,\delta'}$.

After coloring the first t vertices, the expected number of new colors in the second phase of the algorithm is $g(\ell_1, \ell_2, \ldots, \ell_t)$, where:

$$g(\ell_1, \ell_2, \ldots, \ell_m) = \sum_{E \in \mathcal{E}} \sum_{b=1}^{c} \sum_{i=1}^{\min\{|E|,\ell\}-1} \Pr[X_{E,b,i\beta_E} = 1 \mid \ell_1, \ell_2, \ldots, \ell_m], \tag{3.3}$$

Algorithm 1 Finds an ℓ-intersperse coloring for $N = (V, \mathcal{E})$.

Requirement: $c \geq 1$.

{Phase One:}
$\ell_1 = 1$.
for $t = 2$ to $|V|$ **do**
  $\ell_t = 1$.
  for $b = 2$ to c **do**
    if $g(\ell_1, \ldots, \ell_{t-1}, \ell) < g(\ell_1, \ldots, \ell_{t-1}, \ell_t)$ **then**
      $\ell_t = b$.
    end if
  end for{at this point, ℓ_t is the color that minimizes $g(\ell_1, \ldots, \ell_{t-1}, \ell_t)$}
end for
tentatively assign ℓ_i as the color of v_i for all $1 \leq i \leq |V|$.
{Phase Two:}
for all $E \in \mathcal{E}$ **do**
  for $b = 1$ to c **do**
    $C_{E,b}$ = set of vertices of E with color b.
    if $|C_{E,b}| > \beta_E(|E| - 1)$ **then**
      partition $C_{E,b}$ into $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil$ disjoint sets with sizes at most $\beta_E(|E|-1)$.
      assign a different color to each of the first $\left\lceil \frac{|C_{E,b}|}{\beta_E(|E|-1)} \right\rceil - 1$ subsets.
    end if
  end for
end for

and $1 \leq m \leq n$ is an integer and $\beta_E = \frac{1}{\min\{|E|,\ell\}-1}$.

We assign a color $1 \leq \ell_{t+1} \leq c$ to v_{t+1} that minimizes $g(\ell_1, \ell_2, \ldots, \ell_t, \ell_{t+1})$. This can be done in polynomial time if we can evaluate $g(\ell_1, \ell_2, \ldots, \ell_t, \ell_{t+1})$ in polynomial time. Note that

$$p^{(\alpha)}_{n,\delta} \quad = \quad \sum_{j=\lfloor \delta(n-1) \rfloor + 1}^{n} \binom{n}{j} \alpha^j (1-\alpha)^{n-j}. \tag{3.4}$$

Since the value of $\binom{n}{j}$ can be computed in time $O(n^3)$ using simple dynamic programming [27], $p^{(\alpha)}_{n,\delta}$ can be computed in polynomial time, using (3.4). Therefore, we can compute the conditional probabilities in Equation 3.3 in polynomial time. Hence, we can evaluate $g(\ell_1, \ell_2, \ldots, \ell_t, \ell_{t+1})$ in polynomial time, too.

The deterministic algorithm for finding an ℓ-intersperse coloring of N is shown in Algorithm 1. After coloring all n vertices, $g(\ell_1, \ell_2, \ldots, \ell_n)$ is exactly the number of new colors in the second phase. Also, for all $2 \leq t \leq n$, ℓ_t minimizes $g(\ell_1, \ell_2, \ldots, \ell_{t-1}, \ell_t)$, as is shown in lines 3 to 8. Since $g(\ell_1, \ell_2, \ldots, \ell_{t-1})$ is the expected number of new colors after coloring the first $t-1$ vertices, it is the average of $g(\ell_1, \ell_2, \ldots, \ell_{t-1}, 1)$, $g(\ell_1, \ell_2, \ldots, \ell_{t-1}, 2)$, …, $g(\ell_1, \ell_2, \ldots, \ell_{t-1}, c)$, and thus, $g(\ell_1, \ell_2, \ldots, \ell_{t-1}) \geq g(\ell_1, \ell_2, \ldots, \ell_{t-1}, \ell_t)$. Consequently, $g(\ell_1) \geq g(\ell_1, \ell_2) \geq \ldots \geq g(\ell_1, \ell_2, \ldots, \ell_n)$. Thus, in the second phase, i.e. lines 11 to 19, the number of new colors, $g(\ell_1, \ell_2, \ldots, \ell_n)$, is at most $g(\ell_1 = 1)$. On the other hand, $g(1)$ is the expected number of new colors in the second phase when all vertices are colored randomly. Therefore, $g(1)$ is equal to Formula 3.2 which is $c \sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|, \frac{i}{\min\{|E|,\ell\}-1}}$. Hence, the total number of colors at the end of the second phase of the algorithm is at most

$$c + g(\ell_1, \ell_2, \ldots, \ell_n) \leq c + g(1) = c + c \sum_{E \in \mathcal{E}} \sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|, \frac{i}{\min\{|E|,\ell\}-1}}.$$

□

Similar to the greedy coloring algorithm for proper coloring of graphs [83], there is a simple greedy algorithm that finds an ℓ-intersperse coloring for hypergraphs of low degree.

Theorem 3.1.6. *There exists a polynomial time algorithm that, for a given hypergraph $N = (V, \mathcal{E})$ and given integer $\ell \geq 1$, computes an ℓ-intersperse coloring with at most*

$$\max_{v \in V} \left\{ \sum_{E \in \mathcal{E} \text{ s.t. } v \in E} (\min\{|E|, \ell\} - 1) \right\} + 1$$

colors.

Proof. Suppose $P = \langle v_1, v_2, \ldots, v_{|V|} \rangle$ is an arbitrary permutation of the vertices of N. We color the vertices of N in $|V|$ steps; in the ith step we color v_i. Moreover, we assign the colors in a way that the following invariant holds at each step, for all hyperedges $E \in \mathcal{E}$:

$$\text{colors}_i(E) + \text{uncolored}_i(E) \quad \geq \quad \min\{|E|, \ell\} \tag{3.5}$$

where $1 \leq i \leq |V|$, and $\text{colors}_i(E)$ and $\text{uncolored}_i(E)$ are the number of different colors assigned to the vertices of E and the number of uncolored vertices of E after the ith step, respectively. Clearly, if the above invariant holds after the final step for all hyperedges, then the coloring is a valid ℓ-intersperse coloring, because all the vertices are colored and thus any hyperedge $E \in \mathcal{E}$ has at least $\min\{|E|, \ell\}$ different colors.

In addition to preserving Inequality 3.5, we choose the color of each vertex from $\{1, 2, \ldots, k\}$, where

$$k = \max_{v \in V} \left\{ \sum_{E \in \mathcal{E} \text{ s.t. } v \in E} (\min\{|E|, \ell\} - 1) \right\} + 1.$$

We color v_1 with 1. This does not violate Inequality 3.5 for any hyperedge $E \in \mathcal{E}$, because if v_1 is in E, then $\text{colors}_1(E)$ is one and $\text{uncolored}_1(E)$ is $|E| - 1$, and thus, their sum is $|E| \geq \min\{|E|, \ell\}$; otherwise, if v_1 is not in E, then the values of $\text{colors}_1(E)$ and $\text{uncolored}_1(E)$ will be zero and $|E|$, respectively, and thus, their sum is again $|E| \geq \min\{|E|, \ell\}$.

Suppose that the first i vertices of P are assigned to colors from $\{1, 2, \ldots, k\}$ and Inequality 3.5 holds for all hyperedges. To assign a color to v_{i+1}, we consider all hyperedges $\mathcal{F} = \{E_1, E_2, \ldots, E_j\}$ that contain v_{i+1}. Note that $\text{colors}_{i+1}(E) = \text{colors}_i(E)$ and $\text{uncolored}_{i+1}(E) = \text{uncolored}_i(E)$ for hyperedges E outside $\mathcal{F}$. Thus, it is enough to prove that at least one color from $\{1, 2, \ldots, k\}$ can be assigned to v_{i+1} such that $\text{colors}_{i+1}(E) + \text{uncolored}_{i+1}(E) \geq \min\{|E|, \ell\}$ for all $E \in \mathcal{F}$. If for a hyperedge $E \in \mathcal{F}$ we have $\text{colors}_i(E) + \text{uncolored}_i(E) > \min\{|E|, \ell\}$, then, no matter how we color v_{i+1}, Inequality 3.5 will hold for E after step $(i+1)$. However, if $\text{colors}_i(E) + \text{uncolored}_i(E) = \min\{|E|, \ell\}$, we have to assign a color to v_{i+1} that is not used in E before the $(i+1)$th step. Since $\text{colors}_i(E) \leq \min\{|E|, \ell\} - 1$, at most $\sum_{E \in \mathcal{F}} (\min\{|E|, \ell\} - 1)$ colors from $\{1, 2, \ldots, k\}$ cannot be selected as the color of v_{i+1}. Consequently, because $k > \sum_{E \in \mathcal{F}} (\min\{|E|, \ell\} - 1)$, there is at least one color from $\{1, 2, \ldots, k\}$ that can be assigned to v_{i+1} such that Inequality 3.5 holds after the $(i+1)$th step. □

We can simplify the upper bound of Theorem 3.1.6 in terms of the maximum degree of the input hypergraph.

Corollary 3.1.7. *There exists a polynomial time algorithm that, for a given hypergraph $N = (V, \mathcal{E})$ and given integer $\ell \geq 1$, computes an ℓ-intersperse coloring with at most $\ell\Delta(N) - \Delta(N) + 1$ colors.*

Proof. We use Theorem 3.1.6:

$$\begin{aligned}\max_{v\in V}\left\{\sum_{E\in\mathcal{E} \text{ s.t. } v\in E}(\min\{|E|,\ell\}-1)\right\}+1 &\leq \max_{v\in V}\left\{\sum_{E\in\mathcal{E} \text{ s.t. } v\in E}(\ell-1)\right\}+1\\ &= \ell\Delta(N)-\Delta(N)+1.\end{aligned}$$

□

However, if we want to obtain an upper bound on the intersperse chromatic number of hypergraphs as a function of maximum degree, we can do better than Corollary 3.1.7 in many cases, using a more sophisticated method. In Theorem 3.1.10, we use the Lovász Local Lemma [30] (or simply *the Local Lemma*) to obtain an upper bound on the intersperse chromatic number of hypergraphs. A drawback of the Local Lemma, and hence Theorem 3.1.10, is that it is not constructive, i.e. it does not give a polynomial-time deterministic, or even probabilistic, algorithm to find a coloring with the specified number of colors. Below, we state the Local Lemma in a general form:

THEOREM 3.1.8. **The Local Lemma, Asymmetric Case**. *Suppose that* $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ *is a set of events and* $\Gamma(A_i) \subseteq \mathcal{A}$ *is the set of events that are not mutually independent of* A_i*, for all* $1 \leq i \leq n$*. Moreover, suppose there are* n *real numbers* $x_1, x_2, \ldots, x_n \in [0, 1)$ *such that*

$$Pr[A_i] \leq x_i \prod_{A_j\in\Gamma(A_i)}(1-x_j)$$

for all $1 \leq i \leq n$*. Then*

$$Pr[\overline{A_1} \wedge \overline{A_2} \wedge \ldots \wedge \overline{A_n}] > 0.$$

A simple special case of the Local Lemma is the following:

COROLLARY 3.1.9. **Simple Asymmetric Case** *[67]. Let* $\mathcal{A}$ *and* $\Gamma : \mathcal{A} \mapsto 2^{\mathcal{A}}$ *be the same as in Theorem 3.1.8. Then, if*

$$Pr[A_i] \leq \frac{1}{8} \qquad \text{and} \qquad \sum_{A_j\in\Gamma(A_i)} Pr[A_j] \leq \frac{1}{4}$$

for all $1 \leq i \leq n$*, we have*

$$Pr[\overline{A_1} \wedge \overline{A_2} \wedge \ldots \wedge \overline{A_n}] > 0.$$

We use Corollary 3.1.9 to prove the following theorem.

THEOREM 3.1.10. *Suppose $\ell \geq 1$ is an integer and $N = (V, \mathcal{E})$ is a hypergraph with hyperedges of size at least ℓ. Suppose further that each hyperedge in $\mathcal{E}$ has non-empty intersection with at most $f(i)$ hyperedges of size i, and $f(i) > 0$. Then, N has an ℓ-intersperse coloring with*

$$\left\lfloor \max_{\ell \leq i \leq |V|} \left\{ 2(\ell-1) \left(4e^{\ell-1} f(i)\right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1$$

colors.

Proof. We color each vertex of N with a random color picked uniformly from $\{1, 2, \ldots, k\}$, where

$$k = \left\lfloor \max_{\ell \leq i \leq |V|} \left\{ 2(\ell-1) \left(4e^{\ell-1} f(i)\right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1.$$

We use the simple asymmetric case of the Local Lemma to prove that with probability greater than zero this coloring is a valid ℓ-intersperse coloring. Let A_E denote the event that fewer than ℓ different colors are assigned to the vertices of E.

To bound $\Pr[A_E]$ from above, we note that there are $\binom{k}{\ell-1}$ ways of choosing $\ell - 1$ colors and at most $(\ell-1)^{|E|}$ different ways of assigning the chosen colors to the vertices of E. Hence,

$$\Pr[A_E] \leq \binom{k}{\ell-1} \left(\frac{\ell-1}{k}\right)^{|E|}.$$

Since $\binom{a}{b} \leq (\frac{ae}{b})^b$ for any two integers $1 \leq b \leq a$,

$$\Pr[A_E] \leq \left(\frac{ke}{\ell-1}\right)^{\ell-1} \left(\frac{\ell-1}{k}\right)^{|E|} = e^{\ell-1} \left(\frac{\ell-1}{k}\right)^{|E|-\ell+1}.$$

We denote the right-hand side of the above inequality by $g(|E|)$, where

$$g(i) = e^{\ell-1} \left(\frac{\ell-1}{k}\right)^{i-\ell+1}$$

Since A_E is independent of A_F if $E \cap F = \emptyset$, $\Gamma(A_E)$ can be defined as the set of events corresponding to hyperedges that have non-empty intersection with E.

If we prove that $g(i) \le \frac{1}{8}$, for $\ell \le i \le |V|$, and $\sum_{i=\ell}^{|V|} f(i)g(i) \le \frac{1}{4}$, then we can use Corollary 3.1.9 to complete the proof, because

$$\begin{aligned}
\sum_{A_F \in \Gamma(A_E)} \Pr[A_F] &\le \sum_{h=\ell}^{|V|} \sum_{A_F \in \Gamma(A_E) \text{ and } |F|=h} \Pr[A_F] \\
&\le \sum_{h=\ell}^{|V|} \sum_{A_F \in \Gamma(A_E) \text{ and } |F|=h} g(h) \\
&\le \sum_{h=\ell}^{|V|} f(h)g(h).
\end{aligned}$$

Thus, it is enough to consider the following two steps:

1. Proving that $g(i) \le \frac{1}{8}$: since $f(\ell) \ge 1$, we know that $k \ge 8e^{\ell-1}(\ell-1)$. Therefore,

$$g(i) \;=\; e^{\ell-1}\left(\frac{\ell-1}{k}\right)^{i-\ell+1} \le e^{\ell-1}\left(\frac{\ell-1}{8e^{\ell-1}(\ell-1)}\right)^{i-\ell+1} \le \frac{1}{8}$$

2. Proving that $\sum_{i=\ell}^{|V|} f(i)g(i) \le \frac{1}{4}$:

$$\begin{aligned}
\sum_{i=\ell}^{|V|} f(i)g(i) &= \sum_{i=\ell}^{|V|} f(i)e^{\ell-1}\left(\frac{\ell-1}{k}\right)^{i-\ell+1} \\
&\le \sum_{i=\ell}^{|V|} f(i)e^{\ell-1}\left(\frac{\ell-1}{2(\ell-1)\left(4e^{\ell-1}f(i)\right)^{\frac{1}{i-\ell+1}}}\right)^{i-\ell+1} \\
&= \sum_{i=\ell}^{|V|} 2^{\ell-i-3} \\
&< \frac{1}{4}\sum_{j=1}^{\infty} 2^{-j} \\
&= \frac{1}{4}
\end{aligned}$$

□

The following is a simple corollary of Theorem 3.1.10.

COROLLARY 3.1.11. *Suppose $c \geq 0$ is a constant, $\ell \geq 1$ is an integer, and $N = (V, \mathcal{E})$ is a hypergraph with hyperedges of size at least ℓ. If each hyperedge in $\mathcal{E}$ intersects with at most 2^{ci} hyperedges of size i, then*

$$\chi_\ell(N) \leq \left\lfloor 8(\ell-1)e^{\ell-1}2^{c\ell} \right\rfloor + 1.$$

Proof. It suffices to replace $f(i)$ with 2^{ci} in Theorem 3.1.10.

$$\left\lfloor \max_{\ell \leq i \leq |V|} \left\{ 2(\ell-1)\left(4e^{\ell-1}2^{ci}\right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1 \quad \leq \quad \left\lfloor 8(\ell-1)e^{\ell-1}2^{c\ell} \right\rfloor + 1.$$

□

It is worth mentioning that some effort has been made to derandomize the Local Lemma [15, 67, 28, 73]. These methods derandomize the Local Lemma if a number of additional stronger conditions hold. Both Beck's method [15], which is the first constructive version of the Local Lemma, and Molly and Reed's method [67] assume that the outcome of each event A_i is determined by a set of independent random trials $T_i \subseteq \{t_1, t_2, \ldots, t_m\}$. This assumption holds in our case: in Theorem 3.1.10, the outcome of each event A_E is determined by the color of the vertices of E, and the color of each vertex is chosen uniformly from a domain set. But both methods assume that the domain of each random trial t_j has a constant size, which is not true in our case: the domain of each random trial in Theorem 3.1.10 is of size k, which can depend on n.

Czumaj and Scheideler [28] and Salavatipour [73] generalized Beck's method. However, they assume that the domain of each random trial has logarithmic size, which, again, is not true in our case. Therefore, a stronger constructive version of the Local Lemma is required to derandomize Theorem 3.1.10.

Although the results of this section give good upper bounds for some hypergraphs, like bounded degree hypergraphs or hypergraphs with few hyperedges, they do not guarantee any approximation ratio. In the next section, we develop a simple approximation algorithm for the problem of proper coloring of non-uniform hypergraphs.

3.2 Approximation Algorithms

Approximation algorithms for graphs and hypergraphs are studied extensively in the literature. One active area of research is to find approximation algorithms for coloring q-colorable hypergraphs, where q is a constant. Finding a proper coloring for a given hypergraph with a small number of colors is hard even if we know that the given hypergraph is q-colorable: Guruswami et al. [46] showed that there is no polynomial-time algorithm to color a 2-colorable 4-uniform n-vertex hypergraph with $\Omega(\frac{\log\log n}{\log\log\log n})$ colors unless $\mathcal{NP}\subseteq$DTIME$(n^{O(\log\log n)})$. Khot [54] proved a stronger result: for any integer $q \geq 7$ there is a constant c such that it is not possible to color a q-colorable 4-uniform n-vertex hypergraph with $(\log n)^{cq}$ colors, unless $\mathcal{NP}\subseteq$DTIME$(2^{(\lg n)^{O(1)}})$. He also proved that it is $\mathcal{NP}$-hard to color a 3-colorable 3-uniform hypergraph with a constant number of colors [55].

On the other hand, there are a number of positive results on approximate coloring of q-colorable hypergraphs. Krivelevich et al. [58] developed a polynomial-time algorithm to color a 2-colorable 3-uniform n-vertex hypergraph with $O(n^{\frac{1}{5}})$ colors. Also, Alon et al. [8] proved that a 2-colorable n-vertex hypergraph can be colored in polynomial time with $O(n^{1-\frac{1}{r}}\log^{1-\frac{1}{r}} n)$ colors, where r is the rank of the hypergraph.

Note that in the case that the hypergraph is not uniform and can have hyperedges of any size (the rank of the hypergraph can be as big as n), none of the above-mentioned results give a meaningful approximation ratio. The following theorem gives a method to approximately coloring a q-colorable n-vertex hypergraph of any rank.

THEOREM 3.2.1. *For any constant q, there is a polynomial-time algorithm to color an n-vertex hypergraph $N = (V, \mathcal{E})$ with $O(\frac{n}{\lg n})$ colors, if $\chi(N) \leq q$.*

Proof. We know that the vertices of H can be colored with at most q colors such that every hyperedge has at least two different colors. In other words, V can be partitioned into at most q disjoint independent sets. Consequently, every subset $U \subseteq V$ can be partitioned into at most q disjoint independent sets. If the size of U is at most $(q+1)\lg n$, then this partitioning can be found in polynomial time by exhaustive search, because there are at most $q^{|U|} \leq 2^{(q+1)\lg q \lg n} = n^{(q+1)\lg q}$ ways of partitioning U into at most q disjoint subsets.

The algorithm is shown in Figure 2. The algorithm first partitions V into n disjoint subsets of size one. This partitioning is denoted by $\mathcal{C}_0$. Note that $\mathcal{C}_i$ is always a partitioning of V into disjoint independent sets. Also, in lines 3 to 8, at the ith iteration, the algorithm checks if the partitioning $\mathcal{C}_{i-1}$ contains $q+1$ elements $C_1, C_2, \ldots, C_{q+1}$ such that $\sum_{j=1}^{q+1} |C_j| \leq (q+1)\lg n$. There are two cases:

1. For every $q+1$ elements $C_1, C_2, \ldots, C_{q+1} \in \mathcal{C}_{i-1}$, $\sum_{j=1}^{q+1} |C_j| > (q+1)\lg n$: in this case, at most q elements of $\mathcal{C}_{i-1}$ have sizes less than $\lg n$. Thus, because

Algorithm 2 Finds a proper coloring for a q-colorable hypergraph $N = (V, \mathcal{E})$.

```
Set $\mathcal{C}_0 = \{\{v_1\}, \{v_2\}, \ldots, \{v_n\}\}$, where $n$ is the size of $V$ and $v_i$ is the $i$th vertex of $V$.
$i \leftarrow 1$.
**while** $\exists C_1, C_2, \ldots, C_{q+1} \in \mathcal{C}_{i-1}$ such that $\sum_{j=1}^{q+1} |C_j| \leq (q+1)\lg n$ **do**
    $U \leftarrow \cup_{j=1}^{q+1} C_j$.
    partition $U$ into $q$ disjoint independent sets $D_1, D_2, \ldots, D_q$.
    $\mathcal{C}_i = (\mathcal{C}_{i-1} - \{C_1, C_2, \ldots, C_{q+1}\}) \cup \{D_1, D_2, \ldots, D_q\}$.
    $i \leftarrow i + 1$.
**end while**
output $\mathcal{C}_{i-1}$.
```

$\sum_{C \in \mathcal{C}_{i-1}} |C| = n$, $\mathcal{C}_{i-1}$ has at most $\frac{n}{\lg n} + q$ elements: at most $\frac{n}{\lg n}$ elements of size at least $\lg n$, and at most q elements of size less than $\lg n$. Hence, the algorithm has found a partitioning of V into at most $\frac{n}{\lg n} + q$ independent sets. The algorithm terminates at this point.

2. There are $q + 1$ elements $C_1, C_2, \ldots, C_{q+1} \in \mathcal{C}_{i-1}$ such that $\sum_{j=1}^{q+1} |C_j| \leq (q+1)\lg n$: in this case, the algorithm partitions $\cup_{j=1}^{q+1} C_j$ into q independent sets $D_1, D_2, \ldots, D_q$ in time $O(n^{(q+1)\lg q})$. Then, $\mathcal{C}_i$ is built by removing $C_1, C_2, \ldots, C_{q+1}$ from $\mathcal{C}_{i-1}$ and adding the new q independent sets $D_1, D_2, \ldots, D_q$.

Note that, in the second case, $\mathcal{C}_i$ has one element less than $\mathcal{C}_{i-1}$. Therefore, after at most $n - q$ iterations, the condition of the first case will be satisfied and the algorithm will output an $O(\frac{n}{\lg n})$-coloring of N. □

In fact, Theorem 3.2.1 proves a slightly stronger result:

THEOREM 3.2.2. *For any constant q and any n-vertex hypergraph N, there is an algorithm with running-time $O(n^{(q+1)\lg q+2})$ that either finds a coloring of N with $\frac{n}{\lg n} + q$ colors, or proves that N is not q-colorable.*

3.3 Concluding Remarks

We obtained a number of upper bounds for the intersperse coloring problem on general hypergraphs. The technique used in the proof of the main theorem of this chapter, Theorem 3.1.2, was a probabilistic one. We also developed a polynomial-time algorithm based on Theorem 3.1.2. For the special case in which we are seeking a 2-intersperse coloring (i.e. a proper coloring) for an r-uniform hypergraph N, our algorithm uses only $2\left\lceil |\mathcal{E}(N)|^{\frac{1}{r}} \right\rceil$ colors. To the best of our knowledge, this is the first deterministic polynomial-time algorithm that achieves the above-mentioned bound for the proper coloring problem on r-uniform hypergraphs.

Also, we developed a simple $O(\frac{n}{\lg n})$-approximation algorithm for the proper coloring problem on c-colorable general hypergraphs, for any constant c. Before this work, all known results in this area relied on the assumption that the input hypergraphs have a constant rank. On the other hand, we could not generalize our approximation algorithm for the ℓ-intersperse coloring problem.

Chapter 4

Copy Hypergraphs

In this chapter we study the intersperse coloring problem in the family of (induced) copy hypergraphs. As mentioned in Subsection 2.3.3, if N is a $\mathcal{B}$(-induced)-copy hypergraph of A, where A is a graph and $\mathcal{B}$ is a family of connected graphs, N can be presented to an algorithm in two ways: in the full representation in which N itself is given as the input, and in the base representation of N in which a base of N is given as the input.

In this chapter, we first compare the above-mentioned two representations of (induced) copy hypergraphs and argue that the base representation is more interesting to study. Then, in Section 4.2 we study the problem of intersperse coloring of copy hypergraphs when the input is in the base representation. We briefly review previous work on similar graph coloring problems, obtain upper bounds and lower bounds for the intersperse chromatic number, and as a special case of our problem, we develop a property testing algorithm for P_2-freeness.

4.1 Full Representation vs Base Representation

In this subsection, we show that the class of (induced) copy hypergraphs is very similar to the class of general hypergraphs. In particular, we prove that if we are able to solve the problem of computing the ℓ-intersperse chromatic number of N, where N is any copy hypergraph, then we can compute the ℓ-intersperse chromatic number of any hypergraph with an additive error of at most $O(\sqrt{\lg |\{|E| : E \in \mathcal{E}(N)\}|})$. Note that this error is at most $O(\sqrt{\lg |V(N)|})$ in the general case, and is a constant number if N is uniform.

The same claim is true for the case in which a base representation of N, $(A, \mathcal{B})$, is given as the input; however, we can consider more restricted subproblems, such as the case in which only A is a part of the input and $\mathcal{B}$ is fixed. For some classes of graphs $\mathcal{B}$, this subproblem has a nice interpretation in graph theory. We will give some examples in Section 4.2.

THEOREM 4.1.1. *For any hypergraph N and any integer $\ell \geq 1$, there exist a graph A and a family of connected graphs $\mathcal{B}$ such that*

$$\chi_{\ell+t}(O) - t \leq \chi_\ell(N) \leq \chi_{\ell+t}(O),$$

where O is the $\mathcal{B}$-copy hypergraph of A, $t = 5 + 2\left\lceil\sqrt{\frac{1}{4} + 2\lg|X|} + \frac{1}{2}\right\rceil$, and X is the set of sizes of the hyperedges of N; i.e. $X = \{|E| : E \in \mathcal{E}(N)\}$. Furthermore, A and $\mathcal{B}$ can be constructed in polynomial time.

Proof. The idea of the proof is to build A and $\mathcal{B}$ such that the following properties hold.

1. Hypergraphs N and O have the same number of hyperedges.
2. $V(N) \subseteq V(O)$.
3. There exists a bijection b from $\mathcal{E}(N)$ to $\mathcal{E}(O)$ such that for all hyperedges $E_N \in \mathcal{E}(N)$
   (a) $E_N \subseteq b(E_N)$,
   (b) $|b(E_N)| = |E_N| + t$, and
   (c) the t vertices in $b(E_N) - E_N$ are not in any other hyperedge of O.

If we can build A and $\mathcal{B}$ such that the above-mentioned properties hold, then it is easy to see that $\chi_{\ell+t}(O) - t \leq \chi_\ell(N) \leq \chi_{\ell+t}(O)$:

$\chi_\ell(N) \leq \chi_{\ell+t}(O)$: Suppose $c : V(O) \mapsto \{1, 2, \ldots, k\}$ is a valid $(\ell + t)$-intersperse coloring of O. Then, c assigns at least $\min\{|E_O|, \ell + t\}$ different colors to any hyperedge $E_O \in \mathcal{E}(O)$. Therefore, c assigns at least $\min\{|E_O| - t, \ell\} = \min\{|b^{-1}(E_O)|, \ell\}$ different colors to $b^{-1}(E_O)$. Since b is a bijection, we can conclude that c assigns at least $\min\{|E_N|, \ell\}$ different colors to any hyperedge E_N in $\mathcal{E}(N)$.

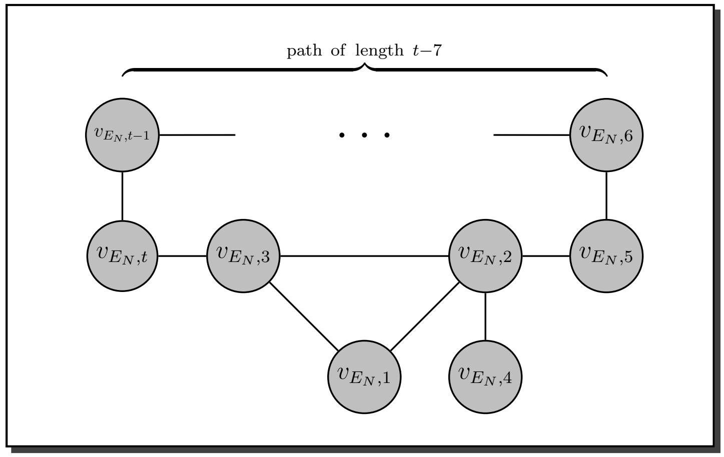

Figure 4.1: The gadget that is added to the hypergraph N for each hyperedge $E_N \in \mathcal{E}(N)$.

$\chi_\ell(N) \geq \chi_{\ell+t}(O) - t$: Suppose $c : V(N) \mapsto \{1, 2, \ldots, k\}$ is a valid ℓ-intersperse coloring of N. Then, consider the coloring c' that assigns $c(v)$ to any vertex $v \in \cup_{E_N \in \mathcal{E}(N)} E_N$, and assigns $k+1$, $k+2$, $\ldots$, $k+t$, in an arbitrary order, to the t vertices in $b(E_N) - E_N$, for any $E_N \in \mathcal{E}(N)$. For any hyperedge $E_O \in \mathcal{E}(O)$, c' assigns $\min\{|b^{-1}(E_O)|, \ell\} = \min\{|E_O| - t, \ell\}$ different colors to $b^{-1}(E_O)$, because $c'(v) = c(v)$ for all vertices in $b^{-1}(E_O)$. Also, c' assigns t different colors to $E_O - b^{-1}(E_O)$. Therefore, c' is a valid ℓ-intersperse coloring of O with $k+t$ colors.

Now we construct A and $\mathcal{B}$. We take the vertices of N as the initial set of vertices of A. Then, for each hyperedge $E_N \in \mathcal{E}(N)$, we add a t-vertex graph A_{E_N}, with vertex set $V(A_{E_N}) = \{v_{E_N,1}, v_{E_N,2}, \ldots, v_{E_N,t}\}$ and edges that are shown in Figure 4.1, to A. In addition to the edges that are shown in Figure 4.1, we add an edge between $v_{E_N,2i+4}$ and $v_{E_N,2j+4}$, for any $1 \leq i < j \leq \left\lceil \sqrt{\frac{1}{4} + 2\lg|X|} + \frac{1}{2} \right\rceil$, if and only if the $\psi_{|X|}(i,j)$th bit of the binary representation of $\phi_N(|E_N|)$ is one, where ψ and ϕ are the following:

$\psi_{|X|}$: This is an arbitrary bijection from the set of all pairs

$$Z(|X|) \quad := \quad \left\{ (i,j) : i,j \in \mathbb{N}, 1 \leq i < j \leq \left\lceil \sqrt{\frac{1}{4} + 2\lg|X|} + \frac{1}{2} \right\rceil \right\}$$

to $\{1, 2, \ldots, |Z(|X|)|\}$. Here, we abuse the notation a bit and use $\psi_{|X|}(i,j)$ instead of $\psi_{|X|}((i,j))$.

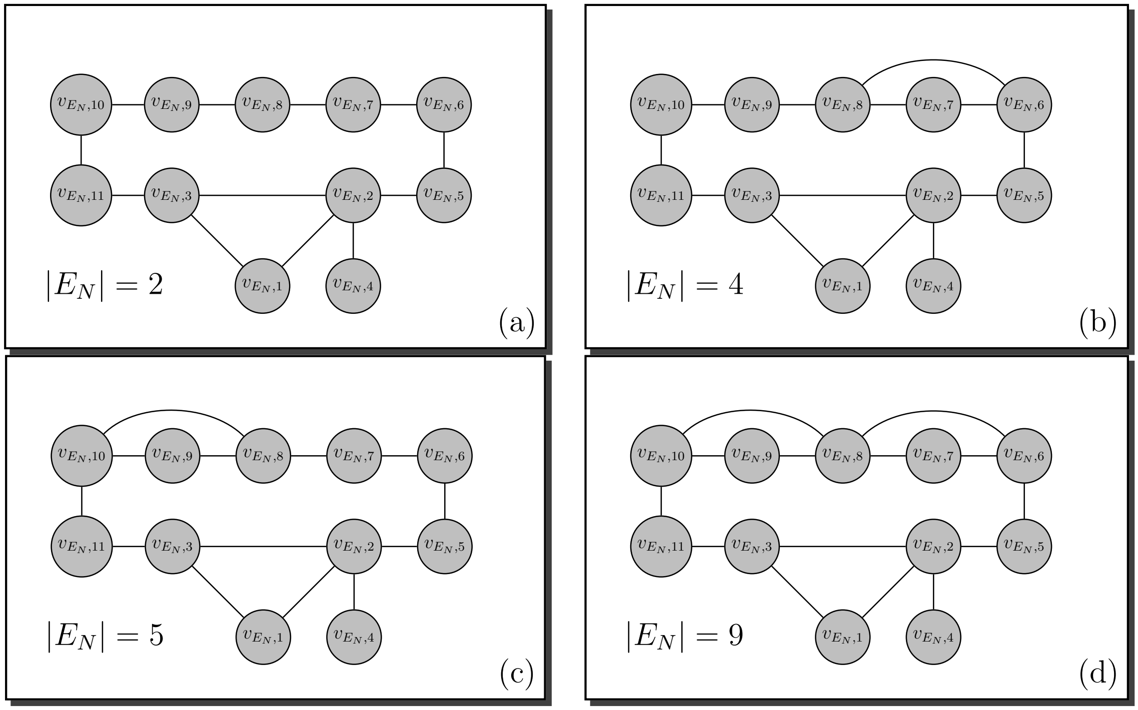

Figure 4.2: If $X = \{2, 4, 5, 9\}$, then A_{E_N} would be of the above graphs, depending on $|E_N|$.

$\phi_N(x)$: This is a function that returns $i - 1$ if x is the ith smallest element of the set X. Therefore, $\phi_N(x)$ is between zero and $|X| - 1$, inclusive.

Moreover, we connect $v_{E_N,1}$ to all vertices in E_N. It is worth to mention that $\left\lceil \sqrt{\frac{1}{4} + 2 \lg |X|} + \frac{1}{2} \right\rceil$ is the smallest value a such that $\binom{a}{2} \geq \lg |X|$. For example, if N has hyperedges of sizes 2, 4, 5, and 9, then A_{E_N}, for a hyperedge $E_N \in \mathcal{E}(N)$, would be one of the graphs shown in Figure 4.2, depending on the size of E_N. In Figure 4.2, we used the following bijection:

$$\psi_4(1,2) = 1 \quad \psi_4(2,3) = 2 \quad \psi_4(1,3) = 3$$

We will show the graphs constructed above have properties 1 and 2 of Lemma 4.1.2. The process of constructing A_{E_N}'s may seem complicated, however, in fact, any polynomial-time construction that has the above-mentioned two properties and uses few vertices can be used in our proof. Below, we prove that our construction has both properties.

Lemma 4.1.2. *If the graphs A_{E_N} are constructed in the above-described method, then,*

1. *For each $E_N^{(1)}, E_N^{(2)} \in \mathcal{E}(N)$, $A_{E_N^{(1)}}$ is isomorphic to $A_{E_N^{(2)}}$ if and only if $E_N^{(1)}$ and $E_N^{(2)}$ are of the same size.*
2. *For each $E-N \in \mathcal{E}(N)$, A_{E_N} is not isomorphic to any subgraph A' of A, unless $A' = A_{E'_N}$ for some hyperedge $E'_N \in \mathcal{E}(N)$.*

Proof.

1. For any $E_N^{(1)}, E_N^{(2)} \in \mathcal{E}(N)$:
   (a) If $A_{E_N^{(1)}}$ is isomorphic to $A_{E_N^{(2)}}$, then $v_{E_N^{(1)},4}$ should be mapped to $v_{E_N^{(2)},4}$, as these are the only degree one vertices. This forces $v_{E_N^{(1)},1}$, $v_{E_N^{(1)},2}$, and $v_{E_N^{(1)},3}$ to be mapped to $v_{E_N^{(2)},1}$, $v_{E_N^{(2)},2}$, and $v_{E_N^{(2)},3}$, respectively. Hence, $v_{E_N^{(1)},i}$ is forced to be mapped to $v_{E_N^{(2)},i}$ for all $1 \leq i \leq t$. Therefore, there is an edge between $v_{E_N^{(1)},2i+4}$ and $v_{E_N^{(1)},2j+4}$ if and only if there is an edge between $v_{E_N^{(2)},2i+4}$ and $v_{E_N^{(2)},2j+4}$, for all $1 \leq i < j \leq \left\lceil \sqrt{\frac{1}{4} + 2\lg|X|} + \frac{1}{2} \right\rceil$. Thus, the $\psi_{|X|}(i,j)$th bit of the binary representation of $\phi_N(|E_N^{(1)}|)$ is one if and only if the $\psi(i,j)$th bit of the binary representation of $\phi_N(|E_N^{(2)}|)$ is one. Consequently, $\phi_N(|E_N^{(1)}|) = \phi_N(|E_N^{(2)}|)$, and hence, $|E_N^{(1)}| = |E_N^{(2)}|$.
   (b) If $|E_N^{(1)}| = |E_N^{(2)}|$, then it is easy to verify that the bijection that maps $v_{E_N^{(1)},i}$ to $v_{E_N^{(2)},i}$, for all $1 \leq i \leq t$, is an isomorphism between $A_{E_N^{(1)}}$ and $A_{E_N^{(2)}}$.
2. For any $E_N \in \mathcal{E}(N)$, if A_{E_N} is isomorphic to a subgraph A' of A, then $V(A')$ should include $v_{E'_N,1}$, $v_{E'_N,2}$, and $v_{E'_N,3}$, where E'_N is a hyperedge in $\mathcal{E}(N)$, and $v_{E_N,1}$, $v_{E_N,2}$, and $v_{E_N,3}$ should be mapped to $v_{E'_N,1}$, $v_{E'_N,2}$, and $v_{E'_N,3}$, respectively, because of the following observation.

   Observation 4.1.3. *Our construction of A has the following two properties:*

   (a) If a vertex $v \in V(A)$ is in a triangle (i.e. a cycle of length three), then $v \in V(A_{E_N})$ for some $E_N \in \mathcal{E}(N)$.
   (b) For any $E_N \in \mathcal{E}(N)$, A_{E_N} has only one triangle with a degree-one neighbor, which is the triangle with vertices $\{v_{E_N,1}, v_{E_N,2}, v_{E_N,3}\}$.

Proof.

(a) Suppose $v \in V(A)$ is not in any $V(A_{E_N})$. Then, v can be connected only to $\left\{v_{E_N^{(1)},1}, v_{E_N^{(2)},1}, \ldots, v_{E_N^{(k)},1}\right\}$, where k is an integer and $E_N^i \in \mathcal{E}(N)$, for $1 \leq i \leq k$. Since there is no edge between $v_{E_N^{(i)},1}$ and $v_{E_N^{(j)},1}$, for any $1 \leq i < j \leq k$, in our construction, v cannot be in any triangle.

(b) This is trivial, because, for any $E_N \in \mathcal{E}(N)$, the only degree-one vertex in A_{E_N} is $v_{E_N,4}$.

□

Similar to the argument of part (a), we can prove that $v_{E_N,i}$ is forced to be mapped to $v_{E'_N,i}$, for all $1 \leq i \leq t$. Therefore, $A' = A_{E'_N}$.

□

Note that all steps of our construction can be done in polynomial time.

Next, we construct $\mathcal{B}$. For any $e \in X$, we add a graph B_e to $\mathcal{B}$. B_e is a graph on $t + e$ vertices $\{v_{e,1}, v_{e,2}, \ldots, v_{e,t}\} \cup \{u_{e,1}, u_{e,2}, \ldots, u_{e,e}\}$, such that,

1. there is no edge between $u_{e,i}$ and $u_{e,j}$ for any $1 \leq i < j \leq e$,
2. $v_{e,1}$ is connected to $u_{e,i}$ for all $1 \leq i \leq e$, and
3. there is an edge between $v_{e,i}$ and $v_{e,j}$ if and only if there is an edge between $v_{E_N,i}$ and $v_{E_N,j}$ for a hyperedge $E_N \in \mathcal{E}(N)$ of size e. Note that only the size of E_N is important here, because, for all hyperedges $E_N^{(1)}, E_N^{(2)} \in \mathcal{E}(N)$ of the same size and all $1 \leq i < j \leq t$, the existence of an edge between $v_{E_N^{(1)},i}$ and $v_{E_N^{(1)},j}$ depends on $\psi_{|X|}(i,j)$ and $\phi_N(|E_N^{(1)}|)$. Similarly, the existence of an edge between $v_{E_N^{(2)},i}$ and $v_{E_N^{(2)},j}$ depends on $\psi_{|X|}(i,j)$ and $\phi_N(|E_N^{(1)}|)$. Therefore, there is an edge between $v_{E_N^{(1)},i}$ and $v_{E_N^{(1)},j}$ if and only if there is an edge between $v_{E_N^{(2)},i}$ and $v_{E_N^{(2)},j}$.

Note that our construction for $\mathcal{B}$ can be completed in polynomial time.

For an example, see Figure 4.3. In Figure 4.3 part (a), the hypergraph N has two hyperedges, of sizes two and three. Hence, $X = \{2, 3\}$, and $Z(|X|) = Z(2) = \{(1, 2)\}$. The only possible bijection ψ_2 from $\{(1, 2)\}$ to $\{1\}$ is $\psi_2(1, 2) = 1$. Also,

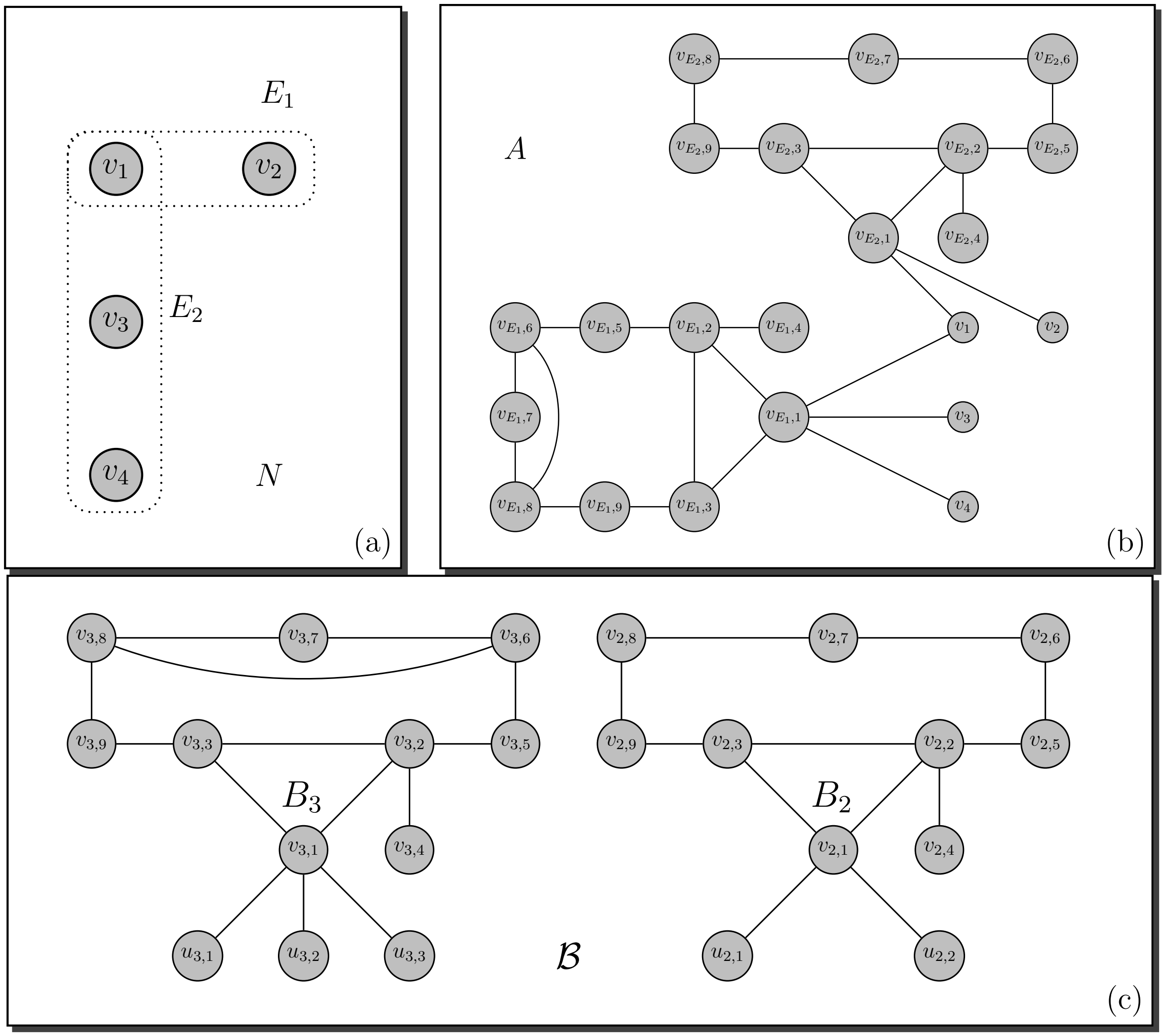

Figure 4.3: An example of constructing A and $\mathcal{B}$ from a hypergraph N.

$\phi_N(2) = 0$ and $\phi_N(3) = 1$, because 2 and 3 are the first and the second smallest elements of X. Using this information, A would be the graph shown in part (b) of Figure 4.3. Also, $\mathcal{B}$ will have two graphs B_2 and B_3 (one for each element of X) that are shown in part (c) of Figure 4.3.

The third property of B_e implies the following observation.

Observation 4.1.4. *The subgraph of B_e induced by $\{v_{e,1}, v_{e,2}, \ldots, v_{e,t}\}$ is isomorphic to A_{E_N} for any E_N of size e.* $\square$

To finish the proof, we show the following three properties hold if O is the $\mathcal{B}$-copy hypergraph of A. These are the three properties, mentioned at the first of this proof, that are sufficient for proving

$$\chi_{\ell+t}(O) - t \leq \chi_{\ell}(N) \leq \chi_{\ell+2}(O).$$

1. $V(N) \subseteq V(A) = V(O)$.

2. By Observation 4.1.4, $B_e[\{v_{e,1}, v_{e,2}, \ldots, v_{e,t}\}]$ is isomorphic to A_{E_N} for any E_N of size e. Also, the second property of Lemma 4.1.2 implies that the only subgraphs of A that are isomorphic to $B_e[\{v_{e,1}, v_{e,2}, \ldots, v_{e,t}\}]$ are A_{E_N}'s for all $E_N \in \mathcal{E}(N)$ of size e. Thus, if a subgraph A' of A is isomorphic to B_e its set of vertices, $V(A')$, should include $V(A_{E_N})$ for some hyperedge $E_N \in \mathcal{E}(N)$ of size e. Since $v_{E_N,1}$ is connected only to the vertices of E in A, $V(A') = E_N \cup V(A_{E_N})$. Therefore, the number of subsets $U \subseteq V(A)$ for which there exists a subgraph A' with $V(A') = U$ such that A' is isomorphic to B_e, is at most the number of subsets $U \subseteq V(A)$ of the form $E_N \cup V(A_{E_N})$, where $E_N \in \mathcal{E}(N)$ is of size e. Also, the number of hyperedges of O is equal to the number of subsets $U \subseteq V(A)$ for which there exists a subgraph A' of A with $V(A') = U$ such that A' is isomorphic to at least one $B_e \in \mathcal{B}$. Hence, the number of hyperedges of O is at most $|\mathcal{E}(N)|$.

   On the other hand, for any $E_N \in \mathcal{E}(N)$, $A[E_N \cup V(A_{E_N})]$ is isomorphic to B_e due to Observation 4.1.4 and the fact that $v_{e,1}$ is connected to $u_{e,i}$ for all $1 \leq i \leq e$. Consequently the number of hyperedges of O is equal to $|\mathcal{E}(N)|$.

3. The bijection $b : \mathcal{E}(N) \mapsto \mathcal{E}(O)$ defined as $b(E_N) = E_N \cup V(A_{E_N})$ has the following properties.

   (a) It is trivial by the definition of b that $E_N \subseteq b(E_N)$ for all $E_N \in \mathcal{E}(N)$.

   (b) By the definition of b, $|b(E_N)| = |E_N \cup V(A_{E_N})|$. Hence, $|b(E_N)| = |E_N| + |V(A_{E_N})| = |E_N| + t$, because A_{E_N} is a t-vertex graph for any hyperedge $E \in \mathcal{E}(N)$.

   (c) The vertices in $b(E_N) - E_N = V(A_{E_N})$ are not in any other hyperedge of O, because all hyperedges E_O of O are of the form $E'_N \cup V(A_{E'_N})$ for some $E' \in \mathcal{E}(N)$, and thus $V(A_{E_N}) \cap E_O = V(A_{E_N}) \cap (E'_N \cup V(A_{E'_N}))$, which is empty, unless $E_N = E'_N$.

□

It is easy to see that the proof of Theorem 4.1.1 works for induced copy hypergraphs without any change, because, in our construction, the graphs in $\mathcal{B}$ are isomorphic only to induced subgraphs of A. Thus, we also have the following theorem.

THEOREM 4.1.5. *For any hypergraph N and any integer $\ell \geq 1$, there exist a graph A and a family of connected graphs $\mathcal{B}$ such that*

$$\chi_{\ell+t}(O) - t \leq \chi_\ell(N) \leq \chi_{\ell+t}(O),$$

where O is the $\mathcal{B}$-induced-copy hypergraph of A, $t = 5 + 2\left\lceil \sqrt{\frac{1}{4} + 2\lg |X|} + \frac{1}{2} \right\rceil$, and $X = \{|E| : E \in \mathcal{E}(N)\}$. Furthermore, A and $\mathcal{B}$ can be constructed in polynomial time.

4.2 Studying the Base Representation

The problem of finding an ℓ-intersperse coloring for a (an induced) copy hypergraph where the hypergraph is given by its base representation can be viewed in a more natural way: for a given graph A and family of connected graphs $\mathcal{B}$, we want to find a coloring of vertices of A such that every (induced) subgraph of A that is isomorphic to a graph $B \in \mathcal{B}$ gets at least $\min\{|V(B)|, \ell\}$ different colors. Moreover, we want to minimize the number of colors. We denote this problem by $\mathrm{SC}(A, \mathcal{B}, \ell)$, and for the induced version we denote the problem by $\mathrm{ISC}(A, \mathcal{B}, \ell)$. Here SC stands for Subgraph Coloring. In Chapter 1 we listed a number of graph coloring problem variants that can be formulated as a special case of $\mathrm{SC}(A, \mathcal{B}, \ell)$.

In the next subsection, we obtain general results for $\mathrm{SC}(A, \mathcal{B}, \ell)$.

4.2.1 General Results

In this section, we first try to apply the general results of Chapter 3 to copy hypergraphs. Theorem 3.1.5 cannot be of much use here, since there is no efficient algorithm to count the number of induced subgraphs of A isomorphic to a graph in $\mathcal{B}$ in the general case.

We apply Corollary 3.1.7 to get a general upper bound on the ℓ-intersperse chromatic number of copy hypergraphs.

Corollary 4.2.1. *There exists a solution for* $\mathrm{SC}(A, \mathcal{B}, \ell)$*, where* A *is a graph and* $\mathcal{B}$ *is a finite family of connected graphs and* $1 \le \ell \le \min_{B \in \mathcal{B}} |V(B)|$ *is an integer, that uses at most*

$$(\ell - 1) \cdot \sum_{B \in \mathcal{B}} \left(|V(B)| \cdot \Delta(A)^{|V(B)|-1} \right) + 1$$

colors.

Proof. Let N be the $\mathcal{B}$-copy hypergraph of A.

Observation 4.2.2. *The maximum degree of* N *is at most the maximum number of subgraphs of* A *that are isomorphic to a graph in* $\mathcal{B}$ *and share a single vertex* $v \in V(A)$.

Proof. Each hyperedge of N corresponds to at least one subgraph of A that is isomorphic to a graph in $\mathcal{B}$. Thus, the number of hyperedges of N that contain v is at most the number of subgraphs of A that are isomorphic to a graph in $\mathcal{B}$ and share a single vertex $v \in V(N)$. □

Suppose v is a vertex in $V(A)$ and c is a one-to-one mapping from $V(B)$ to a subset of $V(A)$ that preserves the edges, where $B \in \mathcal{B}$. Moreover, suppose one vertex of $V(B)$, say x, is mapped to v by c. This can be done in at most $|V(B)|$ different ways. Since c maps any neighbor of x, say y, to a neighbor of v, $c(y)$ can have at most $\Delta(A)$ values for each mapping. After fixing the value of $c(y)$, we can again argue that any neighbor of y should be mapped to a neighbor of $c(y)$ and this can be done in at most $\Delta(A)$ ways. Therefore, since B is connected, after mapping x to v, other vertices of B can be mapped to vertices of A in at most $\Delta(A)^{|V(B)|-1}$ ways. Hence, there are at most $|V(B)| \cdot \Delta(A)^{|V(B)|-1}$ different one-to-one edge-preserving mappings from $V(B)$ to a subset of $V(A)$ that contains v. Thus, we have the following observation.

Observation 4.2.3. A *has at most* $|V(B)| \cdot \Delta(A)^{|V(B)-1|}$ *subgraphs isomorphic to* B *that contain* v.

Observation 4.2.2 and Observation 4.2.3 imply that v is in at most $\sum_{B \in \mathcal{B}} |V(B)| \cdot \Delta(A)^{|V(B)|-1}$ hyperedges of N. Therefore,

$$\Delta(N) \quad \le \quad \sum_{B \in \mathcal{B}} |V(B)| \cdot \Delta(A)^{|V(B)|-1}.$$

Using Corollary 3.1.7 we can conclude that

$$\chi_\ell(N) \quad \le \quad (\ell - 1) \cdot \sum_{B \in \mathcal{B}} \left(|V(B)| \cdot \Delta(A)^{|V(B)|-1} \right) + 1.$$

□

Note that Corollary 4.2.1 is not constructive, as N cannot be constructed from A and B in polynomial time and thus we cannot simply execute the greedy algorithm on N. In some cases, where ℓ is small, Corollary 3.1.11 gives a better upper bound.

COROLLARY 4.2.4. *Suppose A is a graph, $\mathcal{B}$ is a finite family of connected graphs, and $1 \leq \ell \leq \min_{B \in \mathcal{B}} |V(B)|$ is an integer. Then, there exists a solution for $\mathrm{SC}(A, \mathcal{B}, \ell)$ that uses at most*

$$\left\lfloor 2(\ell-1) \max_{\ell \leq i \leq |V(A)|} \left\{ \left(c i b_i \Delta(A)^{i-1} \right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1$$

colors, where $c = 4e^{\ell-1} \max_{B \in \mathcal{B}} |V(B)|$ and $b_i = |\{B : B \in \mathcal{B}, |V(B)| = i\}|$.

Proof. Using Observation 4.2.3, we know that each vertex $v \in V(A)$ is in at most $i\Delta(A)^{i-1}$ subgraphs of A that are isomorphic to $B \in \mathcal{B}$, where $i = |V(B)|$. Since there are b_i graphs of size i in $\mathcal{B}$, we conclude that v is in at most $ib_i\Delta(A)^{i-1}$ subgraphs of A that are isomorphic to a size i graph in $\mathcal{B}$. Hence, if N is the $\mathcal{B}$-copy hypergraph of A, v is in at most $ib_i\Delta(A)^{i-1}$ hyperedges of N of size i. Also, the set of sizes of hyperedges of N is a subset of $\{|B| : B \in \mathcal{B}\}$. Thus, any hyperedge of N is of size at most $\max_{B \in \mathcal{B}} |V(B)|$. Consequently, any hyperedge of N intersects at most

$$f(i) \;=\; ib_i\Delta(A)^{i-1} \max_{B \in \mathcal{B}} |V(B)|$$

other hyperedges of size i. Thus, we can use Theorem 3.1.10 and conclude that

$$\begin{aligned} \chi_\ell(N) &\leq \left\lfloor \max_{\ell \leq i \leq |V(A)|} \left\{ 2(\ell-1) \left(4e^{\ell-1} ib_i \Delta(A)^{i-1} \max_{B \in \mathcal{B}} |V(B)| \right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1 \\ &= \left\lfloor 2(\ell-1) \max_{\ell \leq i \leq |V(A)|} \left\{ \left(c i b_i \Delta(A)^{i-1} \right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1. \end{aligned}$$

□

Below, we give two examples. In the first example, the upper bound of Corollary 4.2.1 is less than the upper bound of Corollary 4.2.4. In the second example, Corollary 4.2.4 gives a better upper bound than Corollary 4.2.1.

Example 4.2.5. Suppose A is an arbitrary graph and $\mathcal{B}$ consists of a number of x-vertex graphs, where $x > 1$ is an integer. Let $y = |\mathcal{B}|$ and N be the $\mathcal{B}$-copy hypergraph of A. Then, using Corollary 4.2.1 we have

$$\begin{aligned}\chi_x(N) &\leq (x-1)\sum_{B\in\mathcal{B}}(x\Delta(A)^{x-1})+1\\ &= y(x-1)x\Delta(A)^{x-1}+1.\end{aligned}$$

However, using Corollary 4.2.4 we will get the following upper bound:

$$\begin{aligned}\chi_x(N) &\leq 2(x-1)\max_{x\leq i\leq |V(A)|}\left\{\left(cib_i\Delta(A)^{i-1}\right)^{\frac{1}{i-x+1}}\right\}+1\\ &= 2(x-1)\left(cxy\Delta(A)^{x-1}\right)+1,\end{aligned}$$

where $c = 4e^{x-1}x$. Thus,

$$\begin{aligned}\chi_x(N) &\leq 2(x-1)\left(cxy\Delta(A)^{x-1}\right)+1\\ &= 8e^{x-1}y(x-1)x^2\Delta(A)^{x-1}+1.\end{aligned}$$

Therefore, Corollary 4.2.1 gives a better upper bound in this example.

Example 4.2.6. Similar to Example 4.2.5, suppose A is an arbitrary graph and $\mathcal{B}$ consists of a number of x-vertex graphs, where $x > 1$ is an integer. Let $y = |\mathcal{B}|$ and N be the $\mathcal{B}$-copy hypergraph of A.

Then, using Corollary 4.2.1 we have

$$\begin{aligned}\chi_2(N) &\leq \sum_{B\in\mathcal{B}}(x\Delta(A)^{x-1})+1\\ &= yx\Delta(A)^{x-1}+1.\end{aligned}$$

However, using Corollary 4.2.4 we will get the following upper bound:

$$\begin{aligned}\chi_2(N) &\leq 2\max_{2\leq i\leq |V(A)|}\left\{\left(cib_i\Delta(A)^{i-1}\right)^{\frac{1}{i-1}}\right\}+1\\ &= 2\left(cxy\right)^{\frac{1}{x-1}}\Delta(A)+1,\end{aligned}$$

where $c = 4ex$. The last equality is due to the fact that b_i is y if $i = x$ and is 0 otherwise. Thus,

$$\begin{aligned}\chi_2(N) &\leq 2\left(cxy\right)^{\frac{1}{x-1}}\Delta(A)+1\\ &= 2\left(4ex^2y\right)^{\frac{1}{x-1}}\Delta(A)+1.\end{aligned}$$

Because every term in $2\left(4ex^2y\right)^{\frac{1}{x-1}}\Delta(A)+1$ is less than $yx\Delta(A)^{x-1}+1$ except the constants, Corollary 4.2.4 gives a better upper bound in this example if we set x to an integer greater than 6.

As was stated in Corollary 2.2.4, the greedy coloring algorithm can properly color any hypergraph A with at most $\Delta(A)+1$ colors. Also, any proper coloring of A is a solution for $\mathrm{SC}(A,\mathcal{B},2)$, for any family of connected graphs $\mathcal{B}$ that have at least two vertices, because a proper coloring of A assigns at least two different colors to any connected subgraph of A. However, for some families of connected graphs, Corollary 4.2.1 and Corollary 4.2.4 give upper bounds bigger than $\Delta(A)+1$. So, for the case of proper coloring we use a result of Lovász to improve our upper bound.

The following lemma is originally due to Lovász [62].

Lemma 4.2.7. *(Lovász [62]) There is a polynomial-time algorithm that can partition the set of vertices of any input graph $A=(V,\mathcal{E})$ into k disjoint sets $V_1,V_2,\dots,V_k$, where k is an integer $1\le k\le |V|$, such that*

1. $\bigcup_{i=1}^{k} V_i = V$*, and*
2. *for all $1\le i\le k$, $\Delta(A[V_i])\le \left\lfloor \frac{\Delta(A)}{k}\right\rfloor$, where $\Delta(A[V_i])$ is the maximum degree in the induced subgraph $A[V_i]$.*

Corollary 4.2.8. *For any graph A and any finite family of connected graphs $\mathcal{B}$, there is a proper coloring of the $\mathcal{B}$-copy hypergraph of A that uses at most*

$$\left\lfloor \frac{\Delta(A)}{\min_{B\in\mathcal{B}}\Delta(B)}+1\right\rfloor$$

colors. Furthermore, this coloring can be found in polynomial time.

Proof. We use Lemma 4.2.7 to find $k=\left\lfloor \frac{\Delta(A)}{\min_{B\in\mathcal{B}}\Delta(B)}+1\right\rfloor$ disjoint sets $V_1,V_2,\dots,V_k$ with the properties stated in Lemma 4.2.7. Then, we assign the vertices of V_i to the ith color. The maximum degree of $A[V_i]$ is at most

$$\begin{aligned}\left\lfloor \frac{\Delta(A)}{k}\right\rfloor &= \left\lfloor \frac{\Delta(A)}{\left\lfloor \frac{\Delta(A)}{\min_{B\in\mathcal{B}}\Delta(B)}+1\right\rfloor}\right\rfloor \\ &< \left\lfloor \frac{\Delta(A)}{\frac{\Delta(A)}{\min_{B\in\mathcal{B}}\Delta(B)}}\right\rfloor \\ &= \min_{B\in\mathcal{B}}\Delta(B),\end{aligned}$$

for all $1 \leq i \leq k$. Thus, $A[V_i]$ cannot have any subgraph isomorphic to any graph in $\mathcal{B}$. Therefore, our coloring does not assign the same color to the vertices of a subgraph B of A that is isomorphic to a graph in $\mathcal{B}$; otherwise, if all the vertices of B are colored with the ith color, B would be a subgraph of $A[V_i]$, which contradicts the fact that $A[V_i]$ does not have any subgraph isomorphic to any graph in $\mathcal{B}$. □

The upper bound of Corollary 4.2.8 is always better than the upper bounds obtained by Corollary 4.2.1 and Corollary 4.2.4 if $\ell = 2$.

For obtaining lower bounds, we use a generalization of the method used by Fertin et al. [38]. They used a simple counting argument to find lower bounds on the acyclic chromatic number and the star chromatic number of graphs. The central part of their proof is the fact that any induced subgraph $A[U]$ of a graph A that does not have a 3-vertex cycle or a 4-vertex path is acyclic, and hence, has at most $|U| - 1$ edges. However, we cannot use this argument for our purpose. In Theorem 4.2.11, we combine the counting method of Fertin et al. [38] with a result in extremal graph theory by Ajtai et al. [4] to obtain a lower bound for $\mathrm{SC}(A, \mathcal{B}, \ell)$, where $\mathcal{B}$ consists of only trees.

We first explain the result of Ajtai et al. [4] that is used in Theorem 4.2.11. Consider the following problem: for a given family of graphs $\mathcal{B}$, what is the maximum number of edges that an n-vertex graph A can have if A does not have any subgraph isomorphic to a graph in $\mathcal{B}$? This problem is called a Turán-type problem, because Turán was the first who investigated a problem of the above-mentioned form. This problem has received extensive attention in the graph theory literature [4, 9, 19, 22, 26, 33, 32, 41, 79]. One of the first results in this area is due to the paper of Turán [79], in which he proved that if $\mathcal{B} = \{K_p\}$, where K_p is a complete p-vertex graph, then A cannot have more than $(1 - \frac{1}{p-1})\frac{n^2}{2}$ edges. For trees, Erdös and Sós proposed the following conjecture [33].

Conjecture 4.2.9. *(Erdös and Sós [33]) Every n-vertex graph with more than $\frac{(k-1)n}{2}$ edges contains all $(k+1)$-vertex trees as subgraphs.*

Conjecture 4.2.9 is known to be true for a number of special families of graphs [29], but is open in the general case. Using Szemerédi's Regularity Lemma [56, 77], Ajtai et al. [4] proved the following approximation version of Conjecture 4.2.9.

Theorem 4.2.10. *(Ajtai et al. [4]) For every $\varepsilon > 0$ there exists a threshold k_0 such that any n-vertex graph with more than $\frac{(1+\varepsilon)kn}{2}$ edges, where $k \geq k_0$, contains all $(k+1)$-vertex trees as subgraphs.*

We use Theorem 4.2.10 to prove the following theorem.

THEOREM 4.2.11. *For every $\varepsilon > 0$, there exists a threshold k_0 such that the following holds: for any graph A, finite family of trees $\mathcal{B}$, and an integer ℓ, if all trees $B \in \mathcal{B}$ have more than k_0 vertices, then there is no solution for* $\mathrm{SC}(A, \mathcal{B}, \ell)$ *with fewer than*

$$\max_{B \in \mathcal{B}} \frac{2|\mathcal{E}(A)|(\min\{|V(B)|, \ell\} - 2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1$$

colors.

Proof. We choose k_0 so that it satisfies Theorem 4.2.10. Suppose c is a solution for $\mathrm{SC}(A, \mathcal{B}, \ell)$ with color classes $V_1, V_2, \ldots, V_k$, m_1 is the number of monocolored edges of A, $m_2 = |\mathcal{E}(A)| - m_1$ is the number of bicolored edges of A, and $B \in \mathcal{B}$ is a tree. Then, since c should assign at least $\ell' = \min\{|V(B)|, \ell\}$ different colors to each subgraph of A that is isomorphic to B, the subgraph of A induced by $V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}$ cannot have any subgraph isomorphic to B, for any $1 \leq a_1 < a_2 < \ldots < a_{\ell'-1} \leq k$. Therefore, $A[V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}]$ has at most $\frac{(1+\varepsilon)(|V(B)|-1) \cdot \sum_{i=1}^{\ell'-1} |V_{a_i}|}{2}$ edges due to Theorem 4.2.10. Hence,

$$\begin{aligned}
& \sum_{1 \leq a_1 < a_2 < \ldots < a_{\ell'-1} \leq k} |\mathcal{E}(A[V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}])| \\
\leq & \sum_{1 \leq a_1 < a_2 < \ldots < a_{\ell'-1} \leq k} \frac{(1+\varepsilon)(|V(B)|-1) \cdot \sum_{i=1}^{\ell'-1} |V_{a_i}|}{2} \\
= & \frac{(1+\varepsilon)(|V(B)|-1)}{2} \sum_{i=1}^{k} \binom{k-1}{\ell'-2} |V_i| \\
= & \frac{(1+\varepsilon)(|V(B)|-1)}{2} \binom{k-1}{\ell'-2} |V(A)|.
\end{aligned}$$

On the other hand, each monocolored edge is counted $\binom{k-1}{\ell'-2}$ times and each bicolored edge is counted $\binom{k-2}{\ell'-3}$ times in $\sum_{1 \leq a_1 < a_2 < \ldots < a_{\ell'-1} \leq k} |\mathcal{E}(A[V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}])|$, because a monocolored edge with color s will be an edge of $A[V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}]$, where $1 \leq a_1 < a_2 < \ldots < a_{\ell'-1} \leq k$, if and only if $a_i = s$, for some $1 \leq i \leq \ell' - 1$. Hence, the values of $a_1, a_2, \ldots, a_{\ell'-1}$ can be chosen in $\binom{k-1}{\ell'-2}$ ways. Similarly, a bicolored edge with colors $s_1 < s_2$ is an edge of $A[V_{a_1} \cup V_{a_2} \cup \ldots \cup V_{a_{\ell'-1}}]$, where $1 \leq a_1 < a_2 < \ldots < a_{\ell'-3} \leq k$, if and only if $a_i = s_1$ and $a_j = s_2$, for some $1 \leq i < j \leq \ell' - 1$. Hence, the values of $a_1, a_2, \ldots, a_{\ell'-1}$ can be chosen in $\binom{k-2}{\ell'-3}$

ways. Thus,

$$\binom{k-1}{\ell'-2}m_1 + \binom{k-2}{\ell'-3}m_2 = \sum_{1\le a_1<a_2<\ldots<a_{\ell'-1}\le k} |\mathcal{E}(A[V_{a_1}\cup V_{a_2}\cup\ldots\cup V_{a_{\ell'-1}}])| \le \binom{k-1}{\ell'-2}\frac{(1+\varepsilon)(|V(B)|-1)|V(A)|}{2}.$$

Consequently,

$$\frac{(1+\varepsilon)(|V(B)|-1)|V(A)|}{2} \ge m_1 + \frac{\binom{k-2}{\ell'-3}}{\binom{k-1}{\ell'-2}}m_2 = m_1 + \frac{\frac{(k-2)!}{(\ell'-3)!(k-\ell'+1)!}}{\frac{(k-1)!}{(\ell'-2)!(k-\ell'+1)!}}m_2 = m_1 + \frac{\ell'-2}{k-1}m_2,$$

and therefore,

$$k \ge \frac{m_2(\ell'-2)}{\frac{(1+\varepsilon)(|V(B)|-1)|V(A)|}{2} - m_1} + 1.$$

If we assume that $|E(A)| \ge \frac{(1+\epsilon)(|V(B)|-1)|V(A)|}{2}$, we will have $0 \le m_1 \le \frac{1}{2}$. Therefore, the right hand side of the above inequality will be minimized if $m_1 = 0$ and $m_2 = |\mathcal{E}(A)|$. Therefore, there is no solution for $\mathrm{SC}(A,\mathcal{B},\ell)$ with fewer than $\frac{2|\mathcal{E}(A)|(\min\{|V(B)|,\ell\}-2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1$ colors. If $|E(A)| < \frac{(1+\epsilon)(|V(B)|-1)|V(A)|}{2}$, the right hand side is not minimized for $m_1 = 0$. However, in the above-mentioned case, $\frac{2|\mathcal{E}(A)|(\min\{|V(B)|,\ell\}-2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1 \le \min\{V(B),\ell\} - 1$. Therefore, for any graph A, there is no solution for $\mathrm{SC}(A,\mathcal{B},\ell)$ with fewer than $\frac{2|\mathcal{E}(A)|(\min\{|V(B)|,\ell\}-2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1$ colors.

Since this argument is true for any $B \in \mathcal{B}$, we conclude that there is no solution for $\mathrm{SC}(A,\mathcal{B},\ell)$ with fewer than

$$\max_{B\in\mathcal{B}} \frac{2|\mathcal{E}(A)|(\min\{|V(B)|,\ell\}-2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1$$

colors. □

We can rephrase the above bound in terms of the average degree of A: for any graph A, finite family of trees $\mathcal{B}$, and an integer ℓ, there is no solution for $\mathrm{SC}(A,\mathcal{B},\ell)$ with fewer than

$$\max_{B\in\mathcal{B}} \frac{\min\{|V(B)|,\ell\}-2}{(1+\varepsilon)(|V(B)|-1)} \cdot \overline{d}(A) + 1$$

colors, where $\overline{d}(A) = \frac{\sum_{v\in V(A)} d(v)}{|V(A)|}$ is the average degree of A.

Note that if we assume Conjecture 4.2.9 is true, the method of Theorem 4.2.11 gives the following result, because Conjecture 4.2.9 is equivalent to Theorem 4.2.10 if we replace $(1+\varepsilon)k$ with $k-1$ and set k_0 to zero.

COROLLARY 4.2.12. *If Conjecture 4.2.9 is true, then the following holds: for any graph A, finite family of trees $\mathcal{B}$, and an integer ℓ, there is no solution for* $\mathrm{SC}(A,\mathcal{B},\ell)$ *with fewer than*

$$\max_{B\in\mathcal{B}} \tfrac{\min\{|V(B)|,\ell\}-2}{|V(B)|-2} \cdot \overline{d}(A) + 1$$

colors.

It is simple to see that Conjecture 4.2.9 is true if we replace $\frac{(k-1)n}{2}$ with $kn-1$ [56]. Thus, we also have the following corollary.

COROLLARY 4.2.13. *For any graph A, finite family of trees $\mathcal{B}$, and an integer ℓ, there is no solution for* $\mathrm{SC}(A,\mathcal{B},\ell)$ *with fewer than*

$$\max_{B\in\mathcal{B}} \tfrac{\min\{|V(B)|,\ell\}-2}{|V(B)|} \cdot \overline{d}(A) + 2$$

colors.

Proof. The proof is similar to the proof of Theorem 4.2.11; only the following inequalities will be changed.

$$\begin{aligned}
&\sum_{1\le a_1<a_2<\ldots<a_{\ell'-1}\le k} |\mathcal{E}(A[V_{a_1}\cup V_{a_2}\cup\ldots\cup V_{a_{\ell'-1}}])| \\
\le\ & \sum_{1\le a_1<a_2<\ldots<a_{\ell'-1}\le k} |V(B)|\cdot\left(\sum_{i=1}^{\ell'-1}|V_{a_i}|\right)-1 \\
<\ & \left(|V(B)|\sum_{i=1}^{k}\binom{k-1}{\ell'-2}|V_i|\right) \\
=\ & \binom{k-1}{\ell'-2}\cdot|V(B)|\cdot|V(A)|
\end{aligned}$$

Also, we will have

$$\begin{aligned}
\binom{k-1}{\ell'-2}m_1+\binom{k-2}{\ell'-3}m_2 &= \sum_{1\le a_1<a_2<\ldots<a_{\ell'-1}\le k} |\mathcal{E}(A[V_{a_1}\cup V_{a_2}\cup\ldots\cup V_{a_{\ell'-1}}])| \\
&< \binom{k-1}{\ell'-2}\cdot|V(B)|\cdot|V(A)|
\end{aligned}$$

and hence

$$m_1 + \frac{\ell' - 2}{k - 1} m_2 < |V(B)| \cdot |V(A)|.$$

Therefore,

$$k \quad > \quad \frac{m_2(\ell' - 2)}{|V(B)| \cdot |V(A)| - m_1} + 1.$$

Again, the right hand side of the above inequality will be minimized if we set $m_1 = 0$ and $m_2 = |\mathcal{E}(A)|$. So, there is no solution for $\mathrm{SC}(A, \mathcal{B}, \ell)$ with fewer than

$$\max_{B \in \mathcal{B}} \tfrac{\min\{|V(B)|, \ell\} - 2}{|V(B)|} \cdot \overline{d}(A) + 2$$

colors. □

4.2.2 Property Testing

In this subsection, we present a property testing algorithm for the following decision problem:

Input: A graph A.

Question: Is it the case that A has no induced subgraph isomorphic to P_2?

In other words, the question is to find whether the P_2-copy hypergraph of A is empty or not.

This is a special case of the well-known problem of testing B-freeness, which is the problem of deciding whether the input graph has an induced subgraph isomorphic to a graph B or not. This problem is a well-studied problem in classical complexity theory [13], parameterized algorithms [12], property testing [6], and many other areas. Alon et al. [6] and Alon [5] have proved that for any graph B there is a graph property tester for testing B-freeness using a constant number of queries in the adjacency matrix model. However, the adjacency matrix model is not a suitable model for testing B-freeness if the input graph is not dense, i.e. has $o(n^2)$ edges, where n is the number of vertices. In the adjacency matrix model, any non-dense graph is ε-close to B-freeness; that is, because the input graph has $o(n^2) < \varepsilon n^2$ edges, one can make the input graph a B-free graph by removing at most εn^2 edges. For non-dense graphs, we may consider the mixed model or the symmetric model. However, we prove there is no graph property tester for B-freeness in the mixed model with query complexity $o(n^{1/2})$, even for the case in which A is a path of length two. Recall that a query in the mixed model can be one of the following, where u and v are vertices:

1. "Is there an edge between u and v?".
2. "How many edges contain v?".
3. "What is the ith neighbor of v?".

Also, recall that the only difference between the mixed model and the symmetric model in graphs is that in the symmetric model we are able to submit an additional query of the form "what are the two end-vertices of E?", where E is an edge. The following observation gives a characterization of P_2-free graphs. This characterization will be used many times in the rest of this chapter.

Observation 4.2.14. *The class of P_2-free graphs, which are graphs that do not have any path of length two as a subgraph, is exactly the family of graphs that are disjoint unions of cliques.*

Proof. Consider shortest paths between every pair of vertices in the same connected component. If one of the shortest paths has length greater than one, then the graph has a P_2 as an induced subgraph.

On the other hand, if the graph is disjoint union of cliques, it does not contain P_2 as an induced subgraph. □

Theorem 4.2.15. *Every tester algorithm for P_2-freeness has to submit at least $\Omega(n^{1/2})$ queries in the mixed model.*

Proof. The key part of the proof is to construct, for every possible value of $0 < \varepsilon < 1$, a subgraph on $O(\sqrt{n})$ vertices that is ε-far from being P_2-free (other vertices of the graph are isolated vertices). Once we do so, the proof is complete. Since the above-mentioned subgraph, which we call B, has $O(\sqrt{n})$ vertices, if we put B at a random location in an empty n-vertex graph A, i.e. an n-vertex graph without any edges, then a single query involves a vertex of B with probability less than $\frac{1}{c\sqrt{n}}$, where c is a constant. Therefore, a single query will not touch B with probability at least $1-\frac{1}{c\sqrt{n}}$. Hence, with probability at least $(1-\frac{1}{c\sqrt{n}})^q > \frac{1}{2}$, where $q \in o(\sqrt{n})$ is the number of queries, the tester algorithm will not touch any vertex of B, i.e. the tester algorithm will not submit any query that involves a vertex of B. Therefore, the algorithm cannot distinguish between A and an empty graph. However, an empty graph is a P_2-free graph while the constructed graph is ε-far from being P_2-free.

LEMMA 4.2.16. *More than* n *edges need to be inserted or deleted to make* $K_{\sqrt{n}+1,\sqrt{n}+1}$ P_2*-free, where* $K_{a,b}$ *is the graph with vertex set* $X \cup Y$*, edge set* $\{\{x,y\} : x \in X, y \in Y\}$*,* $|X| = a$*, and* $|Y| = b$*.*

Proof. Suppose that we can make $K_{\sqrt{n}+1,\sqrt{n}+1}$ P_2-free by m edge modifications. Furthermore, the graph obtained after these edge modifications is the disjoint union of k cliques, say with sets of vertices $C_1, C_2, \ldots, C_k = X \cup Y$, where X and Y are sets of vertices such that $(X \cup Y, \mathcal{E} = \{\{x,y\} : x \in X, y \in Y\}) = K_{\sqrt{n}+1,\sqrt{n}+1}$. For all $i = 1, 2, \ldots, k$ we define $X_i = C_i \cap X$ and $Y_i = C_i \cap Y$.

In order to transform $(X \cup Y, \mathcal{E})$ to a graph that is the disjoint union of k cliques, say with sets of vertices $C_1, C_2, \ldots, C_k$, we need to delete every edge $\{u, v\}$ such that $u \in C_i$, $v \in C_j$, and $i \neq j$. Also, we must add $\{u, v\}$ if for some $i = 1, 2, \ldots, k$ $u \in C_i$, $v \in C_i$, and $\{u, v\} \notin \mathcal{E}$. Therefore,

$$\begin{aligned}
m &= \text{the number of edge deletions } + \text{ the number of edge insertions} \\
&= |\{\{u,v\} : u \in X_i, v \in Y_j, i \neq j\}| + |\{\{u,v\} : u, v \in X_i \text{ or } u, v \in Y_i\}| \\
&= ((\sqrt{n}+1)^2 - \sum_{i=1}^{k}(|X_i| \cdot |Y_i|)) + \left(\sum_{i=1}^{k} \frac{|X_i|(|X_i|-1)}{2} + \sum_{i=1}^{k} \frac{|Y_i|(|Y_i|-1)}{2}\right) \\
&= (\sqrt{n}+1)^2 + 2 \cdot \left(\sum_{i=1}^{k} \frac{|X_i|^2}{4} + \sum_{i=1}^{k} \frac{|Y_i|^2}{4} - \sum_{i=1}^{k} \frac{(|X_i| \cdot |Y_i|)}{2}\right) - \sum_{i=1}^{k} \frac{|X_i|}{2} - \sum_{i=1}^{k} \frac{|Y_i|}{2} \\
&= (\sqrt{n}+1)^2 + 2\sum_{i=1}^{k} \left(\frac{|X_i|}{2} - \frac{|Y_i|}{2}\right)^2 - \frac{1}{2}(\sum_{i=1}^{k}|X_i| + \sum_{i=1}^{k}|Y_i|) \\
&\geq n + 2\sqrt{n} + 1 - \frac{1}{2}(|X| + |Y|) \\
&= n + \sqrt{n} \\
&> n
\end{aligned}$$

□

Lemma 4.2.16 shows that more than n edge modifications are needed to convert $K_{\sqrt{n}+1,\sqrt{n}+1}$ to a P_2-free graph. On the other hand, $K_{\sqrt{n}+1,\sqrt{n}+1}$ has only $(\sqrt{n}+1)^2 = n + 2\sqrt{n} + 1$ edges. Thus, $K_{\sqrt{n}+1,\sqrt{n}+1}$ is $\frac{n}{n+2\sqrt{n}+1}$-far from P_2-freeness. Now, for any value of $0 < \varepsilon < 1$, we can choose n large enough so that $\frac{n}{n+2\sqrt{n}+1} \geq \varepsilon$. Then,

$K_{\sqrt{n}+1,\sqrt{n}+1}$ would be ε-far from P_2-freeness. Since no algorithm can distinguish between the input graph and an empty graph with probability at least $\frac{1}{2}$ if it submits $o(\sqrt{n})$ queries, and the input graph is ε-far from P_2-freeness, every tester algorithm for P_2-freeness has to submit at least $\Omega(\sqrt{n})$ queries.

□

Now, we consider the special case of B-freeness, in which $B = P_2$, and we develop a tester algorithm with query complexity $O(\text{poly}(1/\varepsilon))$ in the symmetric model, for this special case.

In the rest of this section we will use the properties of P_2-free graphs that will be mentioned in Observation 4.2.17. In Observation 4.2.17, we use $\Gamma_C(\{u_0, u_1\})$ to denote the common neighbors of u_0 and u_1.

OBSERVATION 4.2.17. *If A is P_2-free, then every edge $\{u_0, u_1\}$ of A has the following properties:*

1. $d(u_0) = d(u_1)$.
2. $|\Gamma_C(\{u_0, u_1\})| + 1 = d(u_0)$.

Proof. The proof is simple:

1. $d(u_0) = d(u_1)$: The connected component that contains $\{u_0, u_1\}$ is a clique, and thus, $d(u_0) + 1 = d(u_1) + 1 = |C|$, where C is the set of vertices of the connected component containing $\{u_0, u_1\}$.
2. $|\Gamma_C(\{u_0, u_1\})| + 1 = d(u_0)$: The number of common neighbors of u_0 and u_1 is equal to $|C| - 2$. Thus, $|\Gamma_C(\{u_0, u_1\})| + 2 = d(u_0) + 1 = |C|$.

□

We can prove the converse of Observation 4.2.17: if all edges of a graph have the above-mentioned properties, then the graph is P_2-free.

LEMMA 4.2.18. *If all edges of A satisfy properties* 1 *and* 2 *of Observation 4.2.17, then A is P_2-free.*

Proof. Property 1 and Property 2 ensure that, for any edge $\{u_0, u_1\}$ of A, if u_1 is connected to a vertex v, then u_0 is connected to v, too.

Assume that A has an induced subgraph isomorphic to P_2. Let u_0, u_1, and u_2 be the vertices of the subgraph of A that is isomorphic to P_2, and let u_0 and u_2 be the two ends of the path. Then, $\{u_0, u_1\}$ is an edge of A, but u_1 is connected to u_2 and u_0 is not connected to u_2, which is a contradiction. Thus, A cannot have an induced subgraph isomorphic to P_2. □

Next, we develop a tester algorithm for P_2-freeness with query complexity $O(\text{poly}(1/\varepsilon))$. We use two high-level queries in our algorithm:

1. The first high-level query type is to choose a random edge $\{u_0, u_1\} \in \mathcal{E}$. This is equivalent to choosing a random number $1 \leq i \leq |\mathcal{E}|$ and then submitting two queries: "what are the first and second vertices of the ith edge?".

2. The next high-level query type is to perform a random walk of length k from u. This can be done by submitting $4k$ queries in the symmetric model, because by asking the degree of v, choosing a random number $1 \leq i \leq d(v)$, asking for the ith edge that contains v, and finally asking the first and the second vertices of that edge, we can move to a random neighbor of v.

Shown in Algorithm 3, the tester, is relatively simple: it picks a number of random edges in line 2; for each random edge E, it picks a number of random walks W of length two starting from one end of E in line 8. Then, the tester checks if all the vertices of E and W have the same degree and are connected to each other. Note that this is true in any P_2-free graph, because, by Observation 4.2.14, any walk that is started in a clique cannot go outside the clique and hence all the vertices of E and W have the same degree and are connected to each other. If this condition is false for at least one E and one W, then the tester rejects, because it has found evidence that the input graph is not P_2-free; otherwise, since the tester does not find any evidence that the input graph is not P_2-free, the tester accepts the input graph.

Before giving intuition about the correctness of our tester algorithm, we compute its running time. Constants t_1 and t_2 of the algorithm will be set later in this section. Algorithm 3 looks at t_1 random edges of A in line 2. In line 8, for each random edge $\{u_0, u_1\}$, it picks $2t_2$ random neighbors of a vertex. Also, it asks the degree of at most $2t_2$ vertices in line 9. In addition to these queries, it submits at most $4t_2$ queries of the form "does $\mathcal{E}$ contain $\{u, v\}$?" for each random edge in line 12. Therefore, the above algorithm uses at most $O(t_1 \cdot t_2)$ queries overall.

Algorithm 3 P2-FREENESS-TESTER

```
Requirement: A = (V, E), ε
for i ← 1 to t_1 do
    choose a random edge {u_0, u_1} ∈ E
    if d(u_0) ≠ d(u_1) then
        return false
    end if
    for j ← 1 to t_2 do
        r = random number chosen uniformly from {0, 1}
        P = random walk of length 2 starting from u_r
        if there exists a vertex v in P s.t. d(v) ≠ d(u_r) then
            return false
        end if
        if there exists a vertex v in P s.t. {u_0, v} ∉ E or {u_1, v} ∉ E then
            return false
        end if
    end for
end for
return true
```

We first give intuition about the proof and then the formal proof will be presented. The goal is to check the properties mentioned in Observation 4.2.17. A tester algorithm could randomly pick an edge and check if it has the properties of Observation 4.2.17. However, checking the second property is expensive: it requires that the algorithm probes all the neighbors of u_0 and u_1. To overcome this problem, our algorithm checks a number of random neighbors of u_0 and u_1. If the algorithm finds a neighbor of u_0 that is not a neighbor of u_1, or vice versa, then the algorithm rejects. Thus, if the portion of the neighbors of u_0 that are not in $\Gamma_C(\{u_0, u_1\})$ is high enough, then with high probability the algorithm will find one of those neighbors and reject. For simplicity, we use the notation ${\Gamma^+}_C(\{u_0, u_1\})$ to denote $\Gamma_C(\{u_0, u_1\}) \cup \{u_0, u_1\}$.

DEFINITION 4.2.19. We call $\{u_0, u_1\}$ a *bad edge* if

1. $d(u_0) \neq d(u_1)$ or
2. $|\Gamma_C(\{u_0, u_1\})| + 2 = |{\Gamma^+}_C(\{u_0, u_1\})| < (1 - \alpha) \cdot (d(u_0) + 1)$, where the value of $0 < \alpha < \frac{1}{16}$ will be set later. Note that we could use $d(u_1)$ instead of $d(u_0)$, because if the first condition does not hold $d(u_0) = d(u_1)$.

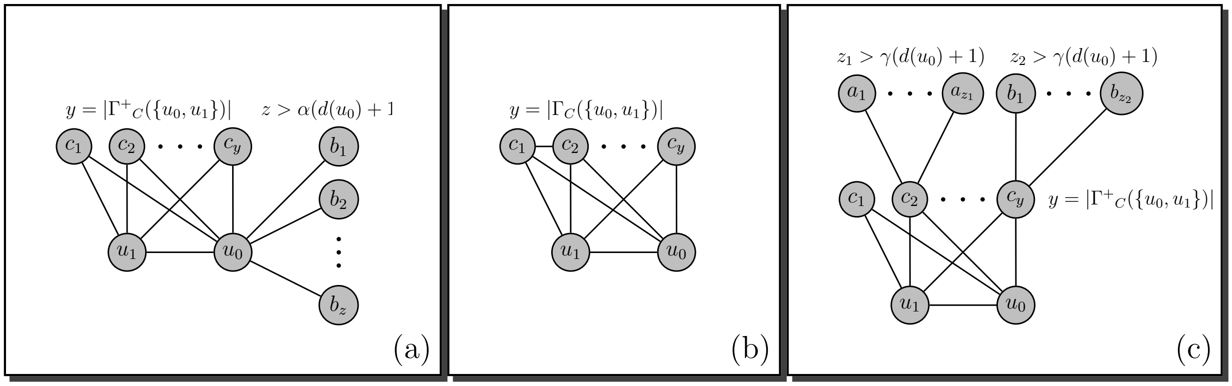

Figure 4.4: A bad edge, good edge, and difficult edge is shown in parts (a), (b), and (c), respectively.

Hence, a bad edge is an edge that violates properties 1 and 2, and this violation can be checked with high probability. An example of a bad edge is shown in Figure 4.4. The edge $\{u_0, u_1\}$ is a bad edge in part (a) of Figure 4.4, because u_0 has more than $\alpha(d(u_0) + 1)$ neighbors outside $\Gamma^+{}_C(\{u_0, u_1\})$. Hence, intuitively, if A is not P_2-free and Algorithm 3 picks a bad edge in line 2, then with high probability the algorithm will reject. Lemma 4.2.22 deals with this case and presents a more precise analysis.

On the other hand, we should prove that if the tester algorithm accepts an input graph A, then A is close to P_2-freeness, i.e. we can convert A to a P_2-free graph by removing or adding at most $\varepsilon|\mathcal{E}|$ edges. Our method of converting A to a P_2-free graph is to repeatedly pick an edge $\{u_0, u_1\}$ from A and make $\Gamma^+{}_C(\{u_0, u_1\})$ a clique by connecting any two vertices from $\Gamma^+{}_C(\{u_0, u_1\})$ that are not connected and removing any edge that is between a vertex in $\Gamma^+{}_C(\{u_0, u_1\})$ and a vertex outside $\Gamma^+{}_C(\{u_0, u_1\})$. We have to be careful in selecting $\{u_0, u_1\}$ so that the method does not require more than $\varepsilon|\mathcal{E}|$ edge modifications. This means that vertices in $\Gamma^+{}_C(\{u_0, u_1\})$ should have few neighbors outside and many neighbors inside $\Gamma^+{}_C(\{u_0, u_1\})$. Thus, we require that $\{u_0, u_1\}$ should not be a bad edge and should have few neighbors that have more than $\gamma \cdot (d(u_0) + 1)$ neighbors outside $\Gamma^+{}_C(\{u_0, u_1\})$. In the following definition, we precisely specify the properties of the edges that we select.

Definition 4.2.20. Suppose $\{u_0, u_1\}$ is an edge, $\{u_0, u_1\}$ is not a bad edge, and v is a vertex. Then, $v \in \Gamma_C(\{u_0, u_1\})$ is called a *bad neighbor* of $\{u_0, u_1\}$ if

1. $d(v) \neq d(u_0)$ or

2. v has more than $\gamma \cdot (d(u_0)+1)$ neighbors outside $\Gamma^{+}{}_{C}(\{u_0, u_1\})$.

We call $\{u_0, u_1\}$ a *good edge* if the following three properties hold:

1. $d(u_0) = d(u_1)$,
2. $|\Gamma^{+}{}_{C}(\{u_0, u_1\})| \geq (1-\alpha) \cdot (d(u_0)+1)$, and
3. the number of bad neighbors of $\{u_0, u_1\}$ is at most $\beta \cdot (d(u_0)+1)$, where the value of $0 < \beta < \frac{1}{16}$ will be set later.

Note that it does not matter if we use $d(u_1)$ instead of $d(u_0)$ in the second property of the definition of good edges, because the first property states that $d(u_0) = d(u_1)$. Also, note that the definitions of bad neighbors and good edges depend on the values of γ and α which will be set later. In a P_2-free graph all edges are good edges, for any values of $0 \leq \alpha, \gamma \leq 1$. An example of a good edge is shown in Figure 4.4. In part (b), $\{u_0, u_1\}$ is a good edge because all three conditions of Definition 4.2.20 hold.

There is still one more case that should be investigated and that is the case in which A has neither many bad edges nor many good edges. Therefore, with high probability our tester algorithm will not find a bad edge and thus will accept A, while, since there are not enough good edges in A, we cannot prove that A is ε-close to P_2-freeness.

Definition 4.2.21. We call any edge that is not a bad edge or a good edge a *difficult* edge. Equivalently, an edge $\{u_0, u_1\}$ is a difficult edge if and only if the following three properties hold:

1. $d(u_0) = d(u_1)$,
2. $|\Gamma^{+}{}_{C}(\{u_0, u_1\})| \geq (1-\alpha) \cdot (d(u_0)+1)$, and
3. the number of bad neighbors of $\{u_0, u_1\}$ is more than $\beta \cdot (d(u_0)+1)$.

The first two properties indicate that $\{u_0, u_1\}$ is not a bad edge, and the third property is for eliminating good edges. An example of a difficult edge is shown in Figure 4.4. In part (c), $\{u_0, u_1\}$ is a difficult edge because it has too many bad neighbors.

In the situation that the numbers of bad edges and good edges are both small, the number of difficult edges is high; thus, with high probability, our tester algorithm will pick a difficult edge in line 2. In Lemma 4.2.23, we prove that if a difficult edge is picked by the tester, the tester will reject A with high probability. Hence, either the tester rejects A or we are able to prove that A is ε-close to P_2-freeness.

Now we formally prove our claims.

LEMMA 4.2.22. *If the edge that is picked by the algorithm in line 2 is a bad edge, then with probability at least* $1-(1-\alpha)^{t_2}$ *the algorithm will reject.*

Proof. If $d(u_0) \neq d(u_1)$, then the algorithm will reject with probability 1 in line 3. Otherwise, in each iteration of the second **for** loop, with probability at least α a vertex outside ${\Gamma^+}_C(\{u_0, u_1\})$ will be chosen as the second vertex of P and the algorithm will reject in line 12. Therefore, with probability at least $1-(1-\alpha)^{t_2}$ our tester algorithm will reject in the second **for** loop. □

LEMMA 4.2.23. *If the edge that is picked by the algorithm in line 2 is a difficult edge, then with probability at least* $1-(1-\beta \cdot \gamma)^{t_2}$ *the algorithm will reject.*

Proof. Again we compute the probability of rejecting the input in each iteration of the second **for** loop. We denote this probability by p_{reject}. Suppose that we denote the three vertices of P by v_0, v_1, and v_2 such that v_0 is either equal to u_0 or equal to u_1, v_1 is adjacent to v_0, and v_2 is adjacent to v_1. The following is a lower bound for p_{reject}.

$$\begin{aligned}
p_{\text{reject}} &\geq \Pr[v_1 \notin \Gamma_C(\{u_0, u_1\})] + \\
&\quad\ \Pr[v_1 \in {\Gamma^+}_C(\{u_0, u_1\}) \text{ and } (d(v_1) \neq d(u_0) \text{ or } v_2 \notin {\Gamma^+}_C(\{u_0, u_1\}))] \\
&= 1 - \frac{|{\Gamma^+}_C(\{u_0, u_1\})|}{d(u_0)} + \\
&\quad\ \frac{|{\Gamma^+}_C(\{u_0, u_1\})|}{d(u_0)} \cdot \Pr[d(v_1) \neq d(u_0) \text{ or } v_2 \notin {\Gamma^+}_C(\{u_0, u_1\}) | v_1 \in {\Gamma^+}_C(\{u_0, u_1\})] \\
&\geq \Pr[d(v_1) \neq d(u_0) \text{ or } v_2 \notin {\Gamma^+}_C(\{u_0, u_1\}) | v_1 \in {\Gamma^+}_C(\{u_0, u_1\})]
\end{aligned}$$

The term in the first inequality is because if v_1 is not in ${\Gamma^+}_C(\{u_0, u_1\})$, then the tester will reject in line 12. Also, if $v_1 \in {\Gamma^+}_C(\{u_0, u_1\})$ but its degree is not equal to $d(u_0)$ or it is not in ${\Gamma^+}_C(\{u_0, u_1\})$, then, again, the tester will reject either in line 9 or line 12.

Since $\{u_0, u_1\}$ is a difficult edge, we know that if $v_1 \in {\Gamma^+}_C(\{u_0, u_1\})$, then v_1 is a bad neighbor of $\{u_0, u_1\}$ with probability at least β. Therefore, with probability at least β either $d(v_1) \neq d(u_0)$ or v_1 has at least $\gamma \cdot (d(v_1)+1)$ neighbors outside ${\Gamma^+}_C(\{u_0, u_1\})$. This shows that

$$\Pr[d(v_1) \neq d(u_0) \text{ or } v_2 \notin {\Gamma^+}_C(\{u_0, u_1\}) | v_1 \in {\Gamma^+}_C(\{u_0, u_1\})] \geq \beta \cdot \gamma,$$

because v_2 is a random neighbor of v_1. Thus, $p_{\text{reject}} \geq \beta \cdot \gamma$. Hence, the algorithm will reject with probability at least $1-(1-\beta\cdot\gamma)^{t_2}$ in the second **for** loop. $\square$

Therefore, if the edge that is picked by the tester algorithm is not a good edge, the tester algorithm will reject with probability at least

$$c = \min\left\{1-(1-\alpha)^{t_2}, 1-(1-\beta\cdot\gamma)^{t_2}\right\}.$$

This leads us to the following theorem.

THEOREM 4.2.24. *If the number of good edges of A is less than $(1-\lambda)\cdot m$, then with probability at least $1-(1-\lambda\cdot c)^{t_1}$ the tester algorithm rejects.*

Proof. In each iteration of the first **for** loop, the algorithm picks a bad edge or a difficult edge with probability at least λ. Therefore, by Lemma 4.2.22 and Lemma 4.2.23, the algorithm will reject with probability at least $\lambda\cdot c$ in each iteration of the first **for** loop. Thus, the algorithm will reject with probability at least $1-(1-\lambda\cdot c)^{t_1}$. $\square$

It remains to prove that if the number of good edges of A is large enough, then A is ε-close to a P_2-free graph.

THEOREM 4.2.25. *If the number of good edges of A is more than $(1-\lambda)\cdot m$, then it is $\left(\frac{\lambda}{2}+\frac{5\alpha+5\beta+\frac{(\alpha+\beta)^2}{(1-2\alpha-2\beta)^2}}{1-8(\alpha+\beta)+(\alpha+\beta)^2}\right)$-close to P_2-freeness.*

Proof. We use Algorithm 4 to convert A to a P_2-free graph using fewer than $\left(\frac{\lambda}{2}+\frac{5\alpha+5\beta+\frac{(\alpha+\beta)^2}{(1-2\alpha-2\beta)^2}}{1-7\alpha-7\beta}\right)\cdot 2m$ edge deletions and edge insertions.

We first explain intuitively how Algorithm 4 works. This algorithm has three main phases.

In the first phase, that is from line 1 to line 15, the algorithm finds a set of good edges $\{u_1,v_1\},\{u_2,v_2\},\ldots,\{u_k,v_k\}$, where k is the final value of i. Also, C_i will be the set of non-bad neighbors of $\{u_i,v_i\}$, because bad neighbors of $\{u_i,v_i\}$ are removed in line 6.

OBSERVATION 4.2.26. *For all $1\leq i\leq k$, the following holds: any vertex that is in $\Gamma^+{}_C(\{u_i,v_i\})$ but is not in C_i is a bad neighbor of $\{u_i,v_i\}$, and vice versa.*

Also, at the ith iteration, in line 8, the algorithm will toss out any good edge that is a non-bad neighbor of $\{u_i, v_i\}$. Moreover, in line 12, the algorithm will toss out any good edge whose non-bad neighbors share a vertex with non-bad neighbors of $\{u_i, v_i\}$. Hence, if a vertex w is in $\Gamma^+{}_C(\{u_i, v_i\})$ and $\Gamma^+{}_C(\{u_j, v_j\})$, then w is a bad neighbor of $\{u_i, v_i\}$, $\{u_j, v_j\}$, or both. In other words, $\Gamma^+{}_C(\{u_i, v_i\})$ and $\Gamma^+{}_C(\{u_j, v_j\})$ are "almost" disjoint, because the number of bad neighbors of a good edge is bounded, by Definition 4.2.20. More precisely, C_i and C_j are completely disjoint, where C_x is $\Gamma^+{}_C(\{u_x, v_x\})$ without bad neighbors of $\{u_x, v_x\}$, for any $1 \leq x \leq k$. Therefore,

OBSERVATION 4.2.27. *For all $1 \leq i < j \leq k$, C_i is disjoint from C_j.*

In the next phase, from line 16 to line 20, any edge that is not completely inside C_i for at least one $1 \leq i \leq k$ is removed from A. Hence, A will be the union of $A[C_i]$'s after line 20. Moreover, since C_i's are mutually disjoint, $A[C_i]$'s are also mutually disjoint, i.e. do not have any vertices or edges in common.

In the final phase, from line 21 to line 25, the algorithm makes every $A[C_i]$ a clique by adding edges inside each $A[C_i]$ if necessary. Therefore, at the end of the algorithm, A will be the union of disjoint cliques. This is formally proved in Lemma 4.2.28.

Algorithm 4 MAKE-P2-FREE

Requirement: $A = (V, \mathcal{E})$
1: $good = \{e : e$ is a good edge$\}$
2: $i = 0$
3: **while** $good \neq \emptyset$ **do**
4: $i = i + 1$
5: select an arbitrary edge $\{u, v\}$ from $good$
6: $C_i = \Gamma^+{}_C(\{u, v\}) - \{$bad neighbors of $\{u, v\}\}$
7: **for all** $\{u', v'\} \in good$ **do**
8: **if** $u' \in C_i$ **or** $v' \in C_i$ **then**
9: $good = good - \{\{u', v'\}\}$
10: **end if**
11: **if** $\exists w \in \Gamma^+{}_C(\{u', v'\}) \cap C_i$ s.t. w is not a bad neighbor of $\{u', v'\}$ **then**
12: $good = good - \{\{u', v'\}\}$
13: **end if**
14: **end for**
15: **end while**
$\{$removing edges between cliques$\}$
16: **for** $\{u, v\} \in \mathcal{E}$ **do**
17: **if** $\nexists i$ s.t. $u \in C_i$ **and** $v \in C_i$ **then**
18: $\mathcal{E} = \mathcal{E} - \{\{u, v\}\}$
19: **end if**
20: **end for**
$\{$inserting edges inside cliques$\}$
21: **for all** $u, v \in V$ s.t. $\{u, v\} \notin \mathcal{E}$ **do**
22: **if** $\exists i$ s.t. $u \in C_i$ **and** $v \in C_i$ **then**
23: $\mathcal{E} = \mathcal{E} \cup \{\{u, v\}\}$
24: **end if**
25: **end for**
26: **return** A

LEMMA 4.2.28. *The output of* MAKE-P2-FREE *is a* P_2*-free graph.*

Proof. Assuming that k is the last value of i in line 4; we prove that the above algorithm transforms the input graph into the union of disjoint cliques with vertex sets $C_1, C_2, \ldots, C_k$.

Because in lines 21–25 every pair of vertices that are in one C_i are connected together, each C_i is a clique in the output graph. Also, in lines 16–20 the algorithm makes sure that there is no other edge in the graph. Thus, the output graph is the union of cliques with vertex sets $C_1, C_2, \ldots, C_k$ and some single vertices.

On the other hand, using Observation 4.2.27, we know that each C_i is disjoint from all C_j's where $1 \leq j < i$. Note that lines 11–12 are necessary, because if we do not remove $\{u', v'\}$ from the set *good* due to w, then $\{u', v'\}$ might be picked by the algorithm in line 5 later in the mth iteration ($m > i$), and since $w \in C_i \cup C_m$, C_i and C_m would not be disjoint.

Considering the above two facts, we can conclude that the output graph is the union of disjoint cliques. □

Now, we calculate the number of edge deletions and edge insertions performed by MAKE-P2-FREE. In order to do this, we divide these modifications into the following groups:

1. R: The set of edges $\{u, v\}$ deleted in lines 16–20 such that neither u nor v is in a C_i.
2. D_i ($1 \leq i \leq k$): The set of edges deleted in lines 16–20 such that one end is in C_i and the other end either is in C_j where $j > i$ or is not in any C_j.
3. I_i ($1 \leq i \leq k$): The set of edges inserted in lines 21–25 such that both ends are in C_i.

Clearly, the number of edge modifications performed by the algorithm is exactly $|R| + \sum_{i=1}^{k} |D_i| + \sum_{i=1}^{k} |I_i|$.

From now on, we set $\alpha = \gamma$. Also, assume $\{u_i, v_i\}$ is the edge picked by the algorithm in line 5 in its ith iteration. Because $\{u_i, v_i\}$ is a good edge and any vertex in C_i is not a bad neighbor, we know that $d(v) = d(u_i)$ for all $v \in C_i$. Thus, we can use d_i to denote the degree of vertices in C_i plus one. In the ideal situation, where the input graph is the union of a number of cliques, C_i would be a clique and d_i would be exactly $|C_i|$. However, in a graph that is close to P_2-freeness, d_i can be larger that $|C_i|$, because there may be some vertices outside C_i that are connected to the vertices inside C_i, like bad neighbors of $\{u_i, v_i\}$.

LEMMA 4.2.29. *There are at most $(\alpha + \beta)d_i^2$ edges with exactly one end in C_i.*

Proof. Since $\{u_i, v_i\}$ is a good edge, u_i and v_i have at most αd_i neighbors outside ${\Gamma^+}_C(\{u_i, v_i\})$ due to the second property of good edges in Definition 4.2.20. Also, the number of bad neighbors of $\{u_i, v_i\}$ is at most βd_i. Therefore, there are at most

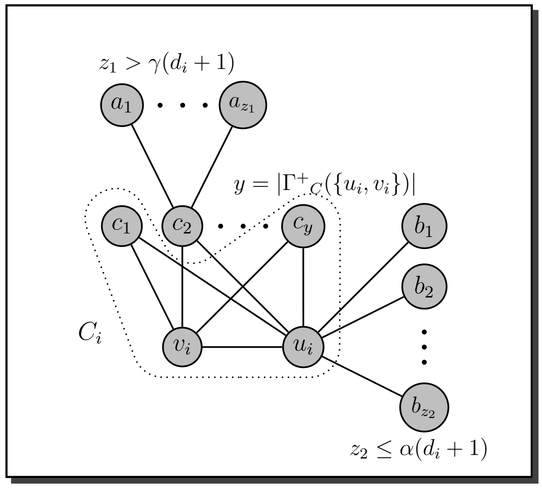

Figure 4.5: Neighbors of $\{u_i, v_i\}$ that are not in C_i.

βd_i vertices that are in $\Gamma^+{}_C(\{u_i, v_i\})$ but not in C_i, according to Observation 4.2.26. For example, in Figure 4.5, the vertices that are in $\Gamma(u_i)$ but not in C_i are $\{c_2\} \cup \{b_1, b_2, \ldots, b_{z_2}\}$, where we know that z_2 is at most $\alpha(d_i + 1)$.

Observation 4.2.30. *u_i and v_i have at most $(\alpha + \beta)d_i$ neighbors outside C_i.*

For a vertex $v \in C_i - \{u_i, v_i\}$, we know that v has at most $\gamma d_i = \alpha d_i$ neighbors outside $\Gamma^+{}_C(\{u_i, v_i\})$, because v is not a bad neighbor of $\{u_i, v_i\}$ (see Observation 4.2.26 and the second property of bad neighbors in Definition 4.2.20). Again, because there are at most βd_i vertices that are in $\Gamma^+{}_C(\{u_i, v_i\})$ but not in C_i, we have the following observation:

Observation 4.2.31. *Any vertex in $C_i - \{u_i, v_i\}$ has at most $(\alpha + \beta)d_i$ neighbors outside C_i.*

Since, due to Observation 4.2.30 and Observation 4.2.31, every vertex in C_i has at most $(\alpha + \beta)d_i$ neighbors outside C_i, there are at most $|C_i|(\alpha + \beta)d_i$ vertices outside C_i that are neighbors to a vertex in C_i. Hence, there are at most $|C_i|(\alpha+\beta)d_i$ edges with exactly one end in C_i. As we mentioned before, $|C_i| \leq d_i$. Hence, we can conclude that there are at most $(\alpha+\beta)d_i^2$ edges with exactly one end in C_i. □

Corollary 4.2.32. *The size of D_i is not greater than $(\alpha + \beta)d_i^2$.*

Proof. D_i is a subset of all edges that have exactly one end in C_i, and the size of the latter set is at most $(\alpha+\beta)d_i^2$, by Lemma 4.2.29. Consequently, $|D_i| \leq (\alpha+\beta)d_i^2$. □

Lemma 4.2.33. *The size of I_i is not greater than $\frac{3}{2}(\alpha+\beta)d_i^2$.*

Proof. Lemma 4.2.29 proves that there are at most $(\alpha+\beta)d_i^2$ edges leaving C_i. Also, because the degree of all vertices in C_i is d_i, we know that there are at least $\frac{1}{2}|C_i|d_i - (\alpha+\beta)d_i^2$ edges inside C_i.

On the other hand, from Observation 4.2.30, we know that u_i has at most $(\alpha+\beta)d_i$ neighbors outside C_i. Therefore, at least $(1-\alpha-\beta)d_i$ neighbors of u_i are in C_i, which proves $|C_i| \geq (1-\alpha-\beta)d_i$. Hence, we can conclude that there are at least $(\frac{1}{2} - \frac{3}{2}(\alpha+\beta))d_i^2$ edges in C_i. Thus, the number of edges that will be inserted in lines 21–25 is at most $\frac{3}{2}(\alpha+\beta)d_i^2$. □

Lemma 4.2.34. *The size of R is not greater than*

$$\lambda m + \left(\frac{(\alpha+\beta)^2}{2(1-2\alpha-\beta)^2}\right)\sum_{i=1}^{k} d_i^2.$$

Proof. Suppose that $\{u,v\} \in R$. Then, either $\{u,v\}$ is not a good edge, or there exists a vertex $w \in {\Gamma^+}_C(\{u,v\})$ such that w is not a bad neighbor of $\{u,v\}$ and $w \in C_i$ for some i. Otherwise, $\{u,v\}$ would be picked by the algorithm in line 5. Let R_g be the subset of all good edges of R. Since, due to the assumption of Theorem 4.2.25, the number of good edges is more than $(1-\lambda)m$, the number of edges that are not good is at most λm. Hence, $|R| \leq \lambda m + |R_g|$. Therefore, by bounding the size of R_g from above we will obtain an upper bound on the size of R.

The key observation to bound $|R_g|$ is that because for every edge $\{u,v\} \in R_g$ there is a vertex $w_{\{u,v\}} \in {\Gamma^+}_C(\{u,v\})$ and a number $i_{\{u,v\}}$ such that $w_{\{u,v\}}$ is not a bad neighbor of $\{u,v\}$ and $w_{\{u,v\}} \in C_{i_{\{u,v\}}}$, both u and v have a lot of neighbors in $C_{i_{\{u,v\}}}$. To see this, we consider the following facts:

1. $d(u) = d(v) = d(w_{\{u,v\}}) = d_{i_{\{u,v\}}} - 1$: Because $\{u,v\}$ is a good edge (see the first property of good edges in Definition 4.2.20), and $w_{\{u,v\}}$ is not a bad neighbor of $\{u,v\}$ (see the first property of bad neighbors in Definition 4.2.20), we have $d(u) = d(v) = d(w_{\{u,v\}})$. Also, since $w_{\{u,v\}}$ is in $C_{i_{\{u,v\}}}$, $d(w_{\{u,v\}}) = d_{i_{\{u,v\}}} - 1$.
2. $w_{\{u,v\}}$ has at least $(1-\alpha)d_{i_{\{u,v\}}}$ neighbors in ${\Gamma^+}_C(\{u,v\})$: Because $w_{\{u,v\}}$ is not a bad neighbor of $\{u,v\}$, it has at most $\gamma(d(u)+1) = \alpha d_{i_{\{u,v\}}}$ neighbors outside ${\Gamma^+}_C(\{u,v\})$.

3. $w_{\{u,v\}}$ has at least $(1-\alpha-\beta)d_{i_{\{u,v\}}}$ neighbors in $C_{i_{\{u,v\}}}$: This fact follows directly from Observation 4.2.31.

OBSERVATION 4.2.35. *For any $\{u,v\} \in R_g$, both u and v are in $T_{i_{\{u,v\}}}$, where T_i is the set of vertices in $V(A) - C_i$ that have at least $(1-2\alpha-\beta)d_i$ neighbors in C_i.*

Proof. Suppose S_1 is the set of neighbors of $w_{\{u,v\}}$ in $\Gamma^+{}_C(\{u,v\})$ and S_2 is the set of neighbors of $w_{\{u,v\}}$ in $C_{i_{\{u,v\}}}$. It is enough to prove that $|S_1 \cap S_2| \geq (1-2\alpha-\beta)d_{i_{\{u,v\}}}$. Since all the vertices in $S_1 \cup S_2$ are neighbors of $w_{\{u,v\}}$, $|S_1 \cup S_2| \leq d(w_{\{u,v\}})$, and by the first fact we have

$$|S_1 \cup S_2| \leq d(w_{\{u,v\}}) \quad < \quad d_{i_{\{u,v\}}}. \tag{4.1}$$

Also, the second and third facts give us the following lower bounds on the size of S_1 and S_2.

$$|S_1| \quad \geq \quad (1-\alpha)d_{i_{\{u,v\}}} \tag{4.2}$$

$$|S_2| \quad \geq \quad (1-\alpha-\beta)d_{i_{\{u,v\}}} \tag{4.3}$$

Since $|S_1 \cap S_2| = |S_1| + |S_2| - |S_1 \cup S_2|$, we can use inequalities 4.1, 4.2, and 4.3 to find a lower bound on the sizes of $S_1 \cap S_2$.

$$\begin{aligned} |S_1 \cap S_2| &\geq (1-\alpha)d_{i_{\{u,v\}}} + (1-\alpha-\beta)d_{i_{\{u,v\}}} - d_{i_{\{u,v\}}} \\ &= (1-2\alpha-\beta)d_{i_{\{u,v\}}} \end{aligned}$$

□

On the other hand, due to Observation 4.2.30 and Observation 4.2.31, every vertex in C_i has at most $(\alpha+\beta)d_i$ neighbors outside C_i. Therefore, we have the following.

$$|T_i| \leq \frac{|C_i| \cdot (\alpha+\beta)d_i}{(1-2\alpha-\beta)d_i} \leq \frac{\alpha+\beta}{1-2\alpha-\beta}|C_i| \leq \frac{\alpha+\beta}{1-2\alpha-\beta}d_i.$$

For an edge $\{u,v\} \in R_g$ with $i_{\{u,v\}} = i$, we proved in Observation 4.2.35 that both u and v are in T_i. Thus, there are at most $\binom{|T_i|}{2} \leq \frac{1}{2}|T_i|^2$ edges $\{u,v\} \in R_g$ with $i_{\{u,v\}} = i$. Hence, for every $1 \leq i \leq k$, there are at most

$$\frac{1}{2}\left(\frac{\alpha+\beta}{1-2\alpha-\beta}d_i\right)^2 = \frac{(\alpha+\beta)^2}{2(1-2\alpha-\beta)^2}d_i^2$$

edges $\{u, v\} \in R_g$ such that $i_{\{u,v\}} = i$. Since $1 \leq i_{\{u,v\}} \leq k$ for any edge $\{u, v\} \in R_g$,

$$|R_g| \leq \sum_{i=1}^{k} \left(\frac{(\alpha+\beta)^2}{2(1-2\alpha-\beta)^2} d_i^2 \right).$$

□

Using Corollary 4.2.32, Lemma 4.2.33, and Lemma 4.2.34, we can conclude that the number of edge modifications, ξ, is

$$\xi = |R| + \sum_{i=1}^{k} |D_i| + \sum_{i=1}^{k} |I_i| \leq \lambda m + \left(\frac{5}{2}\alpha + \frac{5}{2}\beta + \frac{(\alpha+\beta)^2}{2(1-2\alpha-\beta)^2} \right) \sum_{i=1}^{k} d_i^2.$$

Since by ξ edge modifications A is transformed to a P_2-free graph, *PRDist*(A, P_2-freeness) is at most ξ/m, due to Definition 2.4.3. Also, due to Lemma 2.4.4, *SymDist*(A, P_2-freeness) is at least *PRDist*(A, P_2-freeness)/2. Therefore, A is $\frac{\xi}{2m}$-close to P_2-freeness. Also, we have

$$\frac{\xi}{2m} \leq \frac{\lambda}{2} + \frac{1}{4} \left(5\alpha + 5\beta + \frac{(\alpha+\beta)^2}{(1-2\alpha-\beta)^2} \right) \frac{\sum_{i=1}^{k} d_i^2}{m}.$$

Furthermore, $|C_i| \geq 2$ since $\{u_i, v_i\} \subseteq C_i$. Therefore, because the number of edges of the modified graph is at most $m + \sum_{i=1}^{k} |I_i|$, the modified graph is the union of k cliques C_1, C_2, ..., C_k, and $|C_i| \geq 2$ we have

$$\begin{aligned} m + \sum_{i=1}^{k} |I_i| &\geq \sum_{i=1}^{k} \frac{|C_i|(|C_i|-1)}{2} \\ &\geq \sum_{i=1}^{k} \frac{1}{4} |C_i|^2 \\ &\geq \sum_{i=1}^{k} \frac{1}{4} (1-\alpha-\beta)^2 d_i^2. \end{aligned}$$

Therefore, we obtain a lower bound on m:

$$\begin{aligned} m &\geq \frac{1}{4}(1-\alpha-\beta)^2 \sum_{i=1}^{k} d_i^2 - \frac{3}{2}(\alpha+\beta) \sum_{i=1}^{k} d_i^2 \\ &= \frac{1}{4}(1 - 2(\alpha+\beta) + (\alpha+\beta)^2 - 6(\alpha+\beta)) \sum_{i=1}^{k} d_i^2 \\ &= \frac{1}{4}(1 - 8(\alpha+\beta) + (\alpha+\beta)^2) \sum_{i=1}^{k} d_i^2. \end{aligned}$$

Consequently,

$$\frac{\sum_{i=1}^{k} d_i^2}{m} \quad \leq \quad \frac{4}{1-8(\alpha+\beta)+(\alpha+\beta)^2}.$$

Note that, $1-8(\alpha+\beta)+(\alpha+\beta)^2$ and $1-2\alpha-2\beta$ are both non-negative becuase $\alpha, \beta \leq \frac{1}{16}$ by definition. Thus, A is $(\frac{\lambda}{2} + \frac{5\alpha+5\beta+\frac{(\alpha+\beta)^2}{(1-2\alpha-2\beta)^2}}{1-8(\alpha+\beta)+(\alpha+\beta)^2})$-close to P_2-freeness. □

Now, we need to set the values of α, β, γ, λ, t_1 and t_2 such that they depend only on ε. If we choose appropriate values, we can prove the following theorem.

Theorem 4.2.36. *There is a graph property tester for P_2-freeness that uses $O(\frac{1}{\varepsilon^3})$ queries, for $0 < \varepsilon \leq 1$.*

Proof. We set $\lambda = \varepsilon$, $\alpha = \beta = \gamma = \varepsilon/224$, $t_1 = 4/\varepsilon$, and $t_2 = 224^2/\varepsilon^2$. Then, for every graph A:

1. If A is P_2-free, then the algorithm will accept with probability one, because the algorithm rejects only if it finds an induced P_2.

2. If A is ε-far from P_2-freeness, then the number of good edges of A cannot exceed $(1-\varepsilon)m$; otherwise, using Theorem 4.2.25, A is $(\frac{\varepsilon}{2} + \frac{10\alpha+\frac{4\alpha^2}{(1-4\alpha)^2}}{1-16\alpha+4\alpha^2})$-close to P_2-freeness. But, for $0 \leq \varepsilon \leq 1$,

$$\begin{aligned}&\frac{\varepsilon}{2} + \frac{10\alpha+\frac{4\alpha^2}{(1-4\alpha)^2}}{1-16\alpha+4\alpha^2} \leq \frac{\varepsilon}{2} + \frac{10\alpha+\frac{4\alpha}{(1-4\alpha)^2}}{1-16\alpha+4\alpha^2} \leq \frac{\varepsilon}{2} + \frac{\frac{10\alpha}{(1-4\alpha)^2}+\frac{4\alpha}{(1-4\alpha)^2}}{(1-16\alpha)} \\ &\leq \frac{\varepsilon}{2} + \frac{14\alpha}{(1-16\alpha)(1-4\alpha)^2} \leq \frac{\varepsilon}{2} + \frac{\frac{\varepsilon}{16}}{(1-\frac{\varepsilon}{14})(1-\frac{\varepsilon}{56})^2} \leq \frac{\varepsilon}{2} + \frac{\frac{\varepsilon}{16}}{\frac{1}{2}\cdot\left(\frac{1}{2}\right)^2} = \varepsilon.\end{aligned}$$

   Because the number of good edges of A is at most $(1-\varepsilon)m$, we can use Theorem 4.2.24, which states the algorithm will reject with probability at least $1-(1-\varepsilon\cdot c)^{\frac{4}{\varepsilon}}$ where $c = \min\left\{1-(1-\frac{\varepsilon}{224})^{\frac{224^2}{\varepsilon^2}}, 1-(1-\frac{\varepsilon^2}{224^2})^{\frac{224^2}{\varepsilon^2}}\right\}$. Since $c \geq 1/2$, with probability at least $1-(1-\frac{\varepsilon}{2})^{\frac{4}{\varepsilon}} = 1-((1-\frac{\varepsilon}{2})^{\frac{2}{\varepsilon}})^2 \geq 1-e^{-2} \geq 2/3$ the algorithm will reject.

□

4.3 Concluding Remarks

In this chapter, we obtained several upper bounds on the intersperse chromatic number of copy hypergraphs. We also obtained a general lower bound. However, our main focus in obtaining upper and lower bounds was on copy hypergraphs in the general case. For example, Theorem 4.2.11 gives a general lower bound that works for any finite family of trees and any $\ell > 2$. Therefore, it may be possible to optimize the lower bound for more restricted cases.

We also developed a property tester for the problem of checking P_2-freeness. Our tester can check if a graph is P_2-free or is ε-far from being P_2-free using $O(\text{poly}(\frac{1}{\varepsilon}))$ queries. Since the tester algorithm works for both dense and sparse graphs, it shows the power of our proposed property testing model over previously existing property testing models.

Chapter 5

Geometric Hypergraphs Induced by Axis-Parallel Boxes

In this chapter, we consider the proper coloring of geometric hypergraphs induced by axis-parallel k-dimensional boxes. Smorodinsky obtained an upper bound of $O(\lg n)$ for the case that $k = 2$ [76]. However, finding an upper bound better than n remained open in his work. He proposed the following conjecture:

CONJECTURE 5.0.1. *(Smorodinsky [76]) The chromatic number of a hypergraph induced by k-dimensional axis-parallel boxes is at most $O(\lg^{k-1} n)$.*

In this chapter, we find an $o(n)$ upper bound for all values of $k > 2$; in particular, we find an $O(\lg^3 n)$ upper bound for the case that $k = 3$ and $O(n^{1-2^{1-k}} \lg^k n)$ upper bound for other values of k. There is still a big gap between our bounds and Smorodinsky's conjecture.

In order to achieve the mentioned results, we reduce the problem of properly coloring k-dimensional axis-parallel boxes to a purely combinatorial problem, which is the problem of properly coloring k-PC graphs, defined in Definition 2.3.16. Then, we bound the chromatic number of k-PC graphs.

5.1 Reduction to k-PC Graphs

Studying the properties of k-PC graphs has its own theoretical interest. Also, we will show the chromatic number of k-PC graphs is closely related to the chromatic number of geometric hypergraphs induced by axis-parallel boxes.

THEOREM 5.1.1. *Suppose $\mathcal{B}$ is a set of n k-dimensional boxes. Then, we have the following inequality.*

$$\chi(N_{\mathcal{B}}) \ \leq \ (\lfloor \lg n \rfloor + 1)^k \cdot \min\left\{ \mathrm{mcn}(2k, n), (\mathrm{mcn}(k, n))^{2^k} \right\},$$

where $\mathrm{mcn}(k, n)$ *is the maximum chromatic number of any n-vertex k-PC graph.*

Proof. We reduce the problem of coloring k-dimensional boxes to the problem of coloring k-dimensional boxes such that all the boxes share a single point. Then, we bound the chromatic number of any hypergraph induced by k-dimensional boxes that share a single point.

Lemma 5.1.3 breaks down the problem into the above-mentioned easier problem. Intuitively, Lemma 5.1.3 shows that there exists a set of points P and an assignment of the boxes in $\mathcal{B}$ to the points in P such that if a coloring is given for each set of boxes that are assigned to a single point, then the given colorings can be combined to obtain a global coloring by increasing the number of colors by a small factor.

DEFINITION 5.1.2. Suppose $f : P \mapsto \{1, 2, \ldots, m\}$ is a coloring for P, where P is a set of points in $\mathbb{R}^k$ such that $P \cap B \neq \emptyset$ for any $B \in \mathcal{B}$. We call f a $(\mathcal{B}, P, m)$-coloring if it has the following two properties:

1. For every box $B \in \mathcal{B}$, exactly one point p in $P \cap B$ is colored by f_B, where f_B is the minimum of f among all points in $P \cap B$. We say that B *is assigned to* p.

2. If $B_1 \in \mathcal{B}$ and $B_2 \in \mathcal{B}$ intersect, then either B_1 and B_2 are assigned to the same point or $f_{B_1} \neq f_{B_2}$.

Figure 5.1 illustrates an example of a $(\mathcal{B}, P, 3)$-coloring in which $P = \{p_1, p_2, p_3, p_4\}$ and a $\mathcal{B}$ is a set of 2-dimensional boxes. There are six boxes B_1, B_2, ..., B_6 in Figure 5.1, and four points p_1, p_2, p_3, and p_4. The color of each point p_i is written below p_i in the figure. Also, the first condition of Definition 5.1.2 holds, because B_1 and B_2 are assigned to p_1, B_3 and B_4 are assigned to p_3, B_5 is assigned to p_4, and B_6 is assigned to p_2. It can be checked that the second condition holds, too; here we only check two cases. B_2 and B_3 intersect and $f_{B_2} \neq f_{B_3}$, because $f_{B_2} = 2$ and $f_{B_3} = 1$. So the condition is not violated. Also, the condition holds for the intersecting boxes B_3 and B_4, too: they are both assigned to p_3. We leave to the reader the verification that the second condition holds for other pairs of intersecting boxes.

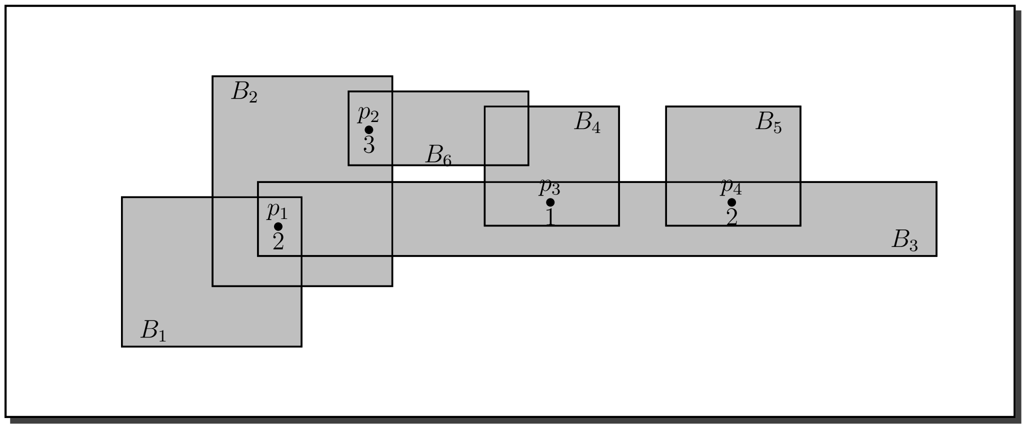

Figure 5.1: A set of points and a coloring that satisfies Properties 1-3.

In the example of Figure 5.1, we could not color p_2 with 1; otherwise, the second condition of Definition 5.1.2 would not hold for B_2 and B_3. Because of the first condition, we could not color p_2 with 2: if we color p_2 with 2, then there are two points in B_2 that are colored with f_{B_2}.

Lemma 5.1.3. *There exists a set of points P and a $(\mathcal{B}, P, g(n,k))$-coloring such that $P \cap B \neq \emptyset$ for all $B \in \mathcal{B}$ and $g(n,k) = (\lfloor \lg n \rfloor + 1)^k)$.*

Proof. We use induction on $k+n$: in the base case, where $k+n=1$, $\mathcal{B}$ consists of only one 0-dimensional box B, because if $n=0$ the lemma becomes trivial. Since a 0-dimensional box is a point, $P = \{B\}$ with the coloring $f(B) = 1$ has both properties of Definition 5.1.2.

Now, we assume $\mathcal{B}$ is a set of n k-dimensional boxes. For each $(k-1)$-dimensional hyperplane Q orthogonal to the first axis, we partition $\mathcal{B}$ into three sets: $\mathcal{B}_Q = \{B : B \cap Q \neq \emptyset\}$, $\mathcal{B}_Q^\ell = \{B : B$ is completely in the left halfspace of $Q\}$, and $\mathcal{B}_Q^r = \{B : B$ is completely in the right halfspace of $Q\}$. Note that a hyperplane Q orthogonal to an axis e can be represented by $e = e(Q)$, where $e(Q) \in \mathbb{R}$ is a constant. Let Q^* be a $(k-1)$-dimensional hyperplane orthogonal to the first axis such that $|\mathcal{B}_{Q^*}^\ell| \leq \lfloor \frac{n}{2} \rfloor$ and $|\mathcal{B}_{Q^*}^r| \leq \lfloor \frac{n}{2} \rfloor$. For example, in Figure 5.2, the dotted line can be Q^*. Below, we show that such a hyperplane always exists. If we move Q from $-\infty$ to $+\infty$ we will have the following two properties.

1. At $-\infty$, $|\mathcal{B}_Q| + |\mathcal{B}_Q^\ell| = 0$.
2. At $+\infty$, $|\mathcal{B}_Q| + |\mathcal{B}_Q^\ell| = n$.

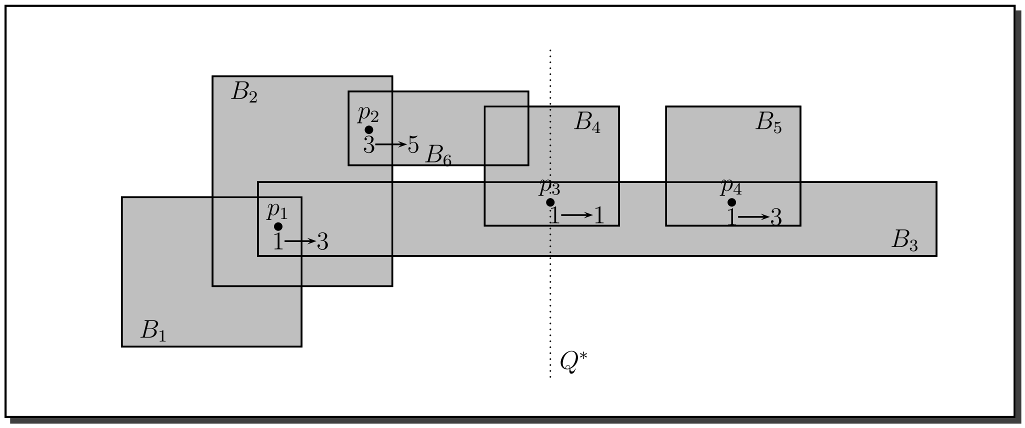

Figure 5.2: At most half of the boxes are completely in the left (right) of Q^*.

We say that a hyperplane Q_1 orthogonal to e is *left* of a hyperplane Q_2 orthogonal to e if $e(Q_1) < e(Q_2)$. Let Q^* be the left-most hyperplane for which $|\mathcal{B}_{Q^*}| + |\mathcal{B}^\ell_{Q^*}| > \lfloor \frac{n}{2} \rfloor$. Since $|\mathcal{B}_Q| + |\mathcal{B}^\ell_Q| \leq \lfloor \frac{n}{2} \rfloor$ for all hyperplanes left of Q^*, $|\mathcal{B}^\ell_{Q^*}| \leq \lfloor \frac{n}{2} \rfloor$. Also, since $|\mathcal{B}^r_{Q^*}| = n - (|\mathcal{B}_{Q^*}| + |\mathcal{B}^\ell_{Q^*}|)$, $|\mathcal{B}^r_{Q^*}| \leq \lfloor \frac{n}{2} \rfloor$.

As the intersection of a k-dimensional box with a $(k-1)$-dimensional hyperplane is a $(k-1)$-dimensional box, the image of $\mathcal{B}_{Q^*}$ on Q^*, i.e. $\{B \cap Q^* : B \in \mathcal{B}_{Q^*}\}$, is a set of at most n $(k-1)$-dimensional boxes. Thus, there exists a set of points P_{Q^*} and a coloring f_{Q^*} which is a $(\text{image}(\mathcal{B}_{Q^*}, Q^*), P_{Q^*}, g(n, k-1))$-coloring, where $\text{image}(\mathcal{B}_Q, Q)$ is the image of $\mathcal{B}_Q$ on Q, i.e. $\text{image}(\mathcal{B}_Q, Q) = \{B \cap Q : B \in \mathcal{B}_Q\}$. Since two boxes $B_1, B_2 \in \mathcal{B}_{Q^*}$ intersect if and only if $B_1 \cap Q^*$ and $B_2 \cap Q^*$ intersect, f_{Q^*} is also a $(\mathcal{B}_{Q^*}, P_{Q^*}, g(n, k-1))$-coloring. In Figure 5.2, $\mathcal{B}_{Q^*} = \{B_3, B_4\}$, $P_{Q^*} = \{p_3\}$, and $f_{Q^*}(p_3) = 1$.

Also, due to the induction hypothesis, there is a set of points $P^\ell_{Q^*}$ and a coloring $f^\ell_{Q^*}$ which is a $(\mathcal{B}^\ell_{Q^*}, P^\ell_{Q^*}, g(\frac{n}{2}, k))$-coloring. Similarly, there is a set of points $P^r_{Q^*}$ and a coloring $f^r_{Q^*}$ which is a $(\mathcal{B}^r_{Q^*}, P^r_{Q^*}, g(\frac{n}{2}, k))$-coloring. In Figure 5.2, we have $\mathcal{B}^\ell_{Q^*} = \{B_1, B_2, B_6\}$, $P^\ell_{Q^*} = \{p_1, p_2\}$, $f^\ell_{Q^*}(p_1) = 1$, and $f^\ell_{Q^*}(p_2) = 3$. For the right halfspace of Q^*, we have $\mathcal{B}^r_{Q^*} = \{B_5\}$, $P^r_{Q^*} = \{p_4\}$, and $f^r_{Q^*}(p_4) = 1$.

We define $P = P_{Q^*} \cup P^\ell_{Q^*} \cup P^r_{Q^*}$. It is clear that $P \cap B \neq \emptyset$ for all $B \in \mathcal{B}$. We verify that the coloring

$$f(p) = \begin{cases} f_{Q^*}(p) & p \in P_{Q^*} \\ g_{Q^*}(n,k) + f^\ell_{Q^*}(p) & p \in P^\ell_{Q^*} \\ g_{Q^*}(n,k) + f^r_{Q^*}(p) & p \in P^r_{Q^*} \end{cases}$$

is a $(\mathcal{B}, P, g(n,k))$-coloring:

1. For every box $B \in \mathcal{B}$, if B is completely in the left halfspace of Q^* (completely in the right halfspace of Q^*), then $B \cap P = B \cap P^{\ell}_{Q^*}$ ($B \cap P = B \cap P^{r}_{Q^*}$) and thus exactly one point in $B \cap P$ is colored by f_B. Otherwise, if $B \cap Q^* \neq \emptyset$, since the colors of all points $p \in P^{\ell}_{Q^*} \cup P^{r}_{Q^*}$ are greater than $g_{Q^*}(n,k)$ and the color of all points $p \in P_{Q^*}$ are at most $g_{Q^*}(n,k)$, again exactly one point in $B \cap P$ is colored by f_B.

2. If $B_1 \in \mathcal{B}$ and $B_2 \in \mathcal{B}$ intersect and are not assigned to the same point, then either both B_1 and B_2 are in $\mathcal{B}_{Q^*}$, $\mathcal{B}^{\ell}_{Q^*}$, or $\mathcal{B}^{r}_{Q^*}$, or one of them is in $\mathcal{B}_{Q^*}$ and the other one is in $\mathcal{B}^{\ell}_{Q^*} \cup \mathcal{B}^{r}_{Q^*}$. In the former case, $f_{B_1} \neq f_{B_2}$ by the induction hypothesis applied to the smaller problem. In the latter case, without loss of generality we assume $B_1 \in \mathcal{B}_{Q^*}$. Then, since $f_{B_1} \leq g_{Q^*}(n,k)$ and $f_{B_2} > g_{Q^*}(n,k)$, f_{B_1} is not equal to f_{B_2}.

Also, $f(p)$ is at most $g(n,k-1) + g(\frac{n}{2},k) = (\lfloor \lg n \rfloor + 1)^{(k-1)} + \lfloor \lg n \rfloor^k \leq g(n,k)$. □

The idea of the rest of the proof is as follows. Suppose P is a set of points such that $P \cap B \neq \emptyset$ for all $B \in \mathcal{B}$ and and f is a $(\mathcal{B}, P, g(n,k))$-coloring. In Lemma 5.1.4, we prove there exists a proper coloring $f_p : \{B \in \mathcal{B} : B \text{ is assigned to } p\} \mapsto \left\{1, 2, \ldots, \min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}\right\}$ of the hypergraph induced by boxes assigned to p. Then, $g : \mathcal{B} \mapsto \{1, 2, \ldots, (\lfloor \lg n \rfloor + 1)^k \cdot \min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}\}$, defined below, is a proper coloring for $N_{\mathcal{B}}$:

$$g(B) \;\; = \;\; (f(p) - 1)(\min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}) + f_p(B), \qquad (5.1)$$

where $p \in P$ is the point to which B is assigned. To prove that g is a proper coloring for $N_{\mathcal{B}}$, consider a point $q \in \cup_{B \in \mathcal{B}} B$. By definition, q will induce a hyperedge $E = \{B \in \mathcal{B} : q \in B\}$. If all boxes $B \in E$ are assigned to a single point p in P, then E is a hyperedge in the hypergraph induced by boxes assigned to p. Hence, since it is a proper coloring, f_p assigns at least two different colors to the elements of E. Consequently, g assigns at least two different colors to the elements of E, because if $f_p(B_1) \neq f_p(B_2)$, then $g(B_1) \neq g(B_2)$ due to Equation 5.1 and the fact that $1 \leq f_p(B) \leq \min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}$. Otherwise, if there exist $B_1, B_2 \in E$ such that B_1 is assigned to p_1 and B_2 is assigned to p_2, Property 2 of Definition 5.1.2 tells us that $f(p_1) \neq f(p_2)$. Therefore, $g(B_1) \neq g(B_2)$ due to the definition of g in Equation 5.1. Hence, g assigns at least two different colors to every hyperedge of $N_{\mathcal{B}}$; therefore, g is a proper coloring.

LEMMA 5.1.4. *The hypergraph induced by boxes assigned to a single point $p \in P$ can be properly colored by at most* $\min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}$ *colors.*

Proof. Suppose $\mathcal{B}_p$ is the set of boxes in $\mathcal{B}$ assigned to p, and, for simplicity, we use N_p to denote $N_{\mathcal{B}_p}$. For all $1 \le i \le k$ we use P_i^r to denote the permutation on $\mathcal{B}_p$ in which $B_1 \in \mathcal{B}_p$ is before $B_2 \in \mathcal{B}_p$ if the right value of the ith range of B_1 is less than the right value of the ith range of B_2. Similarly, for all $1 \le i \le k$ we use P_i^ℓ to denote the permutation on $\mathcal{B}_p$ in which $B_1 \in \mathcal{B}_p$ is before $B_2 \in \mathcal{B}_p$ if the left value of the ith range of B_1 is greater than the left value of the ith range of B_2. In the next paragraph, we will prove that for every hyperedge E of N_p there exist $x_1, x_2, \ldots, x_k$, where $x_i \in \{\ell, r\}$, such that E is $\{P_1^{x_1}, P_2^{x_2}, \ldots, P_k^{x_k}\}$-constructible, and hence, any proper coloring of the graph constructed on $\{P_1^{x_1}, P_2^{x_2}, \ldots, P_k^{x_k}\}$ also assigns at least two different colors to E. Therefore, a coloring that is proper for all $G_{x_1,x_2,\ldots,x_k}$'s, where $G_{x_1,x_2,\ldots,x_k}$ is the graph constructed on $\{P_1^{x_1}, P_2^{x_2}, \ldots, P_k^{x_k}\}$, is a proper coloring for N_p. Because each x_i gets exactly two values, we have exactly 2^k graphs $G_{x_1,x_2,\ldots,x_k}$, and since the $G_{x_1,x_2,\ldots,x_k}$'s are k-PC graphs, a proper coloring with at most $\text{mcn}^{2^k}(k,n)$ colors exists that is proper for all 2^k k-PC graphs $G_{x_1,x_2,\ldots,x_k}$; we just color each k-PC graph separately and compute the final color of a vertex as the 2^k-tuple that consists of its colors in the 2^k k-PC graphs. Also, a proper coloring for the graph constructed on $\left\{P_1^r, P_1^\ell, P_2^r, P_2^\ell, \ldots, P_k^r, P_k^\ell\right\}$ is proper for all $G_{x_1,x_2,\ldots,x_k}$'s, because $\{P_1^{x_1}, P_2^{x_2}, \ldots, P_k^{x_k}\} \subseteq \left\{P_1^r, P_1^\ell, P_2^r, P_2^\ell, \ldots, P_k^r, P_k^\ell\right\}$, for all values of $x_i \in \{\ell, r\}$. Thus a proper coloring with at most $\min\left\{\text{mcn}(2k,n), \text{mcn}^{2^k}(k,n)\right\}$ exists for N_p.

To show that for every hyperedge E of N_p there exist $x_1, x_2, \ldots, x_k$, $x_i \in \{\ell, r\}$, such that E is $\{P_1^{x_1}, P_2^{x_2}, \ldots, P_k^{x_k}\}$-constructible, consider $q \in \mathbb{R}^k$ such that $E = \{B \in \mathcal{B}_p : q \in B\}$. For every $1 \le i \le k$, if the ith coordinate of q is greater than the ith coordinate of p, we set $y_i = r$, otherwise, we set $y_i = \ell$. Therefore, for every $1 \le i \le k$, there exists a number $0 \le a_i \le k$ such that all the first a_i boxes in $P_i^{y_i}$ do not contain q and all the others contain q, because $P_i^{y_i}$ is the sorted list of boxes based on their left values of their ith ranges if the ith coordinate of q is less than the ith coordinate of p; otherwise, $P_i^{y_i}$ is the sorted list of boxes based on their right values of their ith ranges. Hence, due to Definition 2.3.14, E is $\{P_1^{y_1}, P_2^{y_2}, \ldots, P_k^{y_k}\}$-constructible using a_i's as the cut-off points. □

□

Theorem 5.1.1 showed that the chromatic number of k-PC graphs can be used as a upper bound on the chromatic number of $N_{\mathcal{B}}$, where $\mathcal{B}$ is a set of k-dimensional axis-parallel boxes and $N_{\mathcal{B}}$ is the hypergraph induced by $\mathcal{B}$. In the following theorem, we show how we can obtain a lower bound on the chromatic number of $N_{\mathcal{B}}$ based on the chromatic number of k-PC graphs.

THEOREM 5.1.5. *There exists an n-vertex hypergraph N induced by k-dimensional boxes such that $\chi(N) \geq \text{mcn}(k,n)$, where $\text{mcn}(k,n)$ is the maximum chromatic number of any n-vertex k-PC graph.*

Proof. The proof is quite simple: suppose $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ is any set of permutations on a set V of size n. Without loss of generality, we can assume $V = \{1, 2, \ldots, n\}$. We construct a set of n k-dimensional boxes $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ such that B_i is a box with one end in the origin and the opposite end in $(P_1^{-1}[i], P_2^{-1}[i], \ldots, P_k^{-1}[i])$, where $P_j^{-1}[i]$ is the location of i in P_j. More precisely, $B_i = \{(x_1, x_2, \ldots, x_k) : \forall 1 \leq j \leq k \quad x_i \in [0, P_j^{-1}[i]]\}$. Then, it is easy to verify that the chromatic number of the hypergraph induced by $\mathcal{B}$, denoted by N, is at least the chromatic number of the k-PC graph constructed on $\mathcal{P}$: we only consider hyperedges of size two in N. We prove that the set of size-two hyperedges of N are exactly the set of size-two $\mathcal{P}$-constructible sets.

Suppose $\{a, b\} \in \mathcal{E}(N)$. Then, by Observation 2.3.15, it is enough to show that there is no obstacle to $\{a, b\}$: for the sake of contradiction, assume $c \in \{1, 2, \ldots, n\} - \{a, b\}$ is an obstacle to $\{a, b\}$. Since $\{a, b\} \in \mathcal{E}(N)$, there is a point $p = (p_1, p_2, \ldots, p_k)$ in $\mathbb{R}^k$ that is in $B_a \cap B_b$, but not in any other B_i. In particular, $p \notin B_c = \{(x_1, x_2, \ldots, x_k) : \forall 1 \leq j \leq k \quad x_i \in [0, P_j^{-1}[c]]\}$. Hence, there exists $1 \leq j \leq k$ such that $p_j \notin [0, P_j^{-1}[c]]$, and thus, $p_j > P_j^{-1}[c]$. On the other hand, since $p \in B_a \cap B_b$, $p_j \in [0, P_j^{-1}[a]] \cap [0, P_j^{-1}[b]]$, and thus, $p_j \leq P_j^{-1}[a]$ and $p_j \leq P_j^{-1}[b]$. Therefore, neither a nor b is before c in P_j, contradicting with c being an obstacle to $\{a, b\}$.

In fact, the chromatic number of N is exactly the chromatic number of the k-PC graph constructed on $\mathcal{P}$, because for any hyperedge E of size more than two in N there is another hyperedge E' in N of size two such that $E' \subseteq E$. □

5.2 Small k's

We prove that for $1 \leq k \leq 3$, every k-PC graph can be properly colored with a constant number of colors. One of the tools we use to prove the above-mentioned statement is the following lemma in which we show that any subgraph of a k-PC graph can be converted to a k-PC graph by adding some edges (without adding vertices).

Lemma 5.2.1. *Suppose A is a k-PC graph and B is an n-vertex subgraph of A. Then, B is a subgraph of an n-vertex k-PC graph. In particular, B is a subgraph of the graph constructed on $\mathcal{P}[V(B)]$, where $\mathcal{P}$ is the set of k permutations on which A is constructed.*

Proof. Suppose B has an edge $\{u, v\}$ that is not an edge of the graph constructed on $\mathcal{P}[V(B)]$. Due to Definition 2.3.16, $\{u, v\}$ is not $\mathcal{P}[V(B)]$-constructible. Thus, due to Observation 2.3.15, there exists a vertex w that is an obstacle to $\{u, v\}$ in $\mathcal{P}[V(B)]$. However, this means that w is an obstacle to $\{u, v\}$ in $\mathcal{P}$. Therefore, $\{u, v\}$ is not an edge of A, which contradicts the fact that B is a subgraph of A. □

Note that B is not necessarily a k-PC graph itself; for example, if A is any k-PC graph that is not a clique and B is an independent set of A of size $n > 1$, then B is a subgraph of an n-vertex k-PC graph, but is not a k-PC graph itself.

Now, we prove that the chromatic numbers of 1-PC, 2-PC, and 3-PC graphs are each bounded by a constant.

Theorem 5.2.2. *If $k \leq 3$, the chromatic number of any k-PC graph is bounded by a constant. In particular, 1-PC graphs and 2-PC graphs are 2-colorable and 3-PC graphs are 7-colorable.*

Proof. We first consider the cases where $k = 1$ or $k = 2$. A 1-PC graph A has exactly one edge, because, if P_1 is the permutation on which A is constructed, there is only one way of choosing a_1 in Definition 2.3.14 such that U has two elements.

If A is a 2-PC graph, A is 1-degenerate, i.e. A is a tree. To see this, assume that B is an arbitrary n-vertex subgraph of A. Lemma 5.2.1 says that B is a subgraph of an n-vertex 2-PC graph C. Suppose $\{P_1, P_2\}$ is the set of permutations on which C is constructed. By Definition 2.3.16 every edge U of C is $\{P_1, P_2\}$-constructible. In Definition 2.3.14 a_1 can be between 0 and $n-2$, since if a_1 is greater than $n-2$, then U would have fewer than two elements. Also, once we chose a_1, there is only one way of choosing a_2 such that U has two elements. Therefore, we have at most $n - 1$ ways of choosing a_1 and a_2 such that U has two elements. Since B is a subgraph of C and C has at most $n - 1$ edges, B has at most $n - 1$ edges too. Therefore, B must have a vertex with degree at most one, which proves that A is 1-degenerate. Therefore, A is 2-colorable.

The more sophisticated case is needed when $k = 3$. We prove that every 3-PC graph A is 6-degenerate. Assume that B is an arbitrary n-vertex subgraph of A. Using Lemma 5.2.1 we can say B is a subgraph of an n-vertex 3-PC graph $C = (V, \mathcal{E})$. It is enough to prove C has a vertex with degree at most six.

For the sake of contradiction, assume every vertex of C has at least seven neighbors. Also, suppose $\{P_1, P_2, P_3\}$ is the set of permutations on which C is constructed. For every vertex $v \in V$, let $N_i^-(v)$ be the set of all neighbors of v that are before v in P_i. Similarly, let $N_i^+(v)$ be the set of all neighbors of v that are after v in P_i. Let u_1, u_2, and u_3 be the first neighbors of v in P_1, P_2, and P_3; note that u_1, u_2 and u_3 are not necessarily distinct. Since v has more than four neighbors, it has a neighbor w which is not in $\{u_1, u_2, u_3\}$. Because $\{v, w\} \in \mathcal{E}$ does not have any obstacle, every neighbor of v, other than w, appears before both v and w in at least one of the permutations. Similarly, since $\{v, u_1\} \in \mathcal{E}$ does not have any obstacle, w appears before both v and u_1 in at least one of the permutations. Therefore, every neighbor of v appears in at least one $N_i^-(v)$. Suppose that $w \in N_x^-(v)$, where $1 \leq x \leq 3$. Since $\{v, u_x\} \in \mathcal{E}$ does not have any obstacle, and w is after u_x in P_x, w appears before both u_x and v in at least one of the permutations other than P_x. Thus, w is in at least two $N_i^-(v)$s. This argument holds for every neighbor w of v that is not in $\{u_1, u_2, u_3\}$; i.e. every neighbor of v which is not in $\{u_1, u_2, u_3\}$ is in at least two $N_i^-(v)$s. Hence,

$$\sum_{i=1}^{3} |N_i^-(v)| \geq 2d(v) - 3 \Rightarrow \sum_{i=1}^{3} |N_i^+(v)| = 3d(v) - \sum_{i=1}^{3} |N_i^-(v)| \leq d(v) + 3$$

$$\Rightarrow \quad 3|\mathcal{E}| = \sum_{i=1}^{3} \sum_{v \in V} |N_i^+(v)| = \sum_{v \in V} \sum_{i=1}^{3} |N_i^+(v)| \leq \sum_{v \in V} (d(v) + 3) = 2|\mathcal{E}| + 3n \Rightarrow |\mathcal{E}| \leq 3n.$$

However, since every vertex in v has at least seven neighbors, we know $|\mathcal{E}| \geq \frac{7}{2}n$. $\square$

For k-PC graphs with $k \geq 4$, we cannot provide a bound as strong as above. However, we will prove that the chromatic number of all k-PC graphs is in $O(n^{1-\lambda(k)})$, where λ is a function depending only on k.

5.3 A General Upper Bound

All results in Smorodinsky's paper [76] are obtained using the following technique: either he proves that the graph family is d-degenerate for some d, or he reduces the problem to finding the chromatic number of a family of d-degenerate graphs. We applied the same technique to find upper bounds for the chromatic number of k-PC graphs, when k is less than four. However, it seems unlikely that this technique can be used for larger values of k, because, as is shown in Figure 5.3, there are 4-PC graphs with large minimum degree.

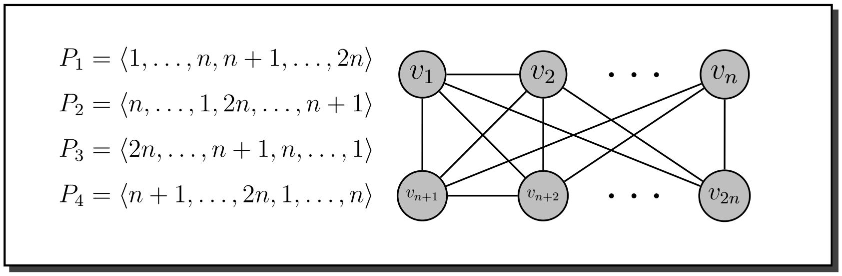

Figure 5.3: A $2n$-vertex 4-PC graph with minimum degree n.

To bound the chromatic number of k-PC graphs, we execute two steps: first we prove each k-PC graph has an independent set of size $O(n^{\lambda(k)})$, where λ is a function of k. Then, we conclude that the chromatic number of k-PC graphs is at most $O(n^{1-\lambda(k)})$.

In order to find a large independent set in k-PC graphs, we use a generalization of a theorem by Erdös and Szekeres [34]. Erdös and Szekeres proved that every sequence of n^2+1 integers either has a non-decreasing subsequence of length $n+1$ or has a non-increasing subsequence of length $n+1$. A non-decreasing or non-increasing sequence is called a *monotonic* sequence. In unpublished work N. G. de Bruijn generalized their result to sequences of k-tuples of numbers [60]. His generalization is stated in the following theorem. More details about De Bruijn's Theorem and other generalizations of Erdös and Szekeres' theorem can be found in papers by Kruskal [60] and Odlyzko et al. [69].

Theorem 5.3.1. *(De Bruijn [60]) Every sequence of $n^{2^k}+1$ k-tuples of integers has a subsequence S of length $n+1$ such that the sequence of the ith elements of k-tuples in S is monotonic for every $1 \leq i \leq k$. Furthermore, $n^{2^k}+1$ is the smallest number that has this property.*

Using De Bruijn's Theorem, we can easily find a lower bound on the size of a maximum independent set of k-PC graphs. Let $\alpha^{PC}(k,n)$ be the maximum number such that every n-vertex k-PC graph has an independent set of size $\alpha^{PC}(k,n)$. In the following lemma, we prove that $\alpha^{PC}(k,n) \in \Omega(n^{\frac{1}{2^{k-1}}})$.

Lemma 5.3.2. *Every n-vertex k-PC graph has an independent set of size $\left\lfloor \frac{1}{2}(n-1)^{\frac{1}{2^{k-1}}} \right\rfloor + 1$.*

Proof. Suppose A is an n-vertex k-PC graph with vertices $\{v_1, v_2, \ldots, v_n\}$. We denote the set of k permutations on which A is constructed by $\mathcal{P} = \{P_1, \ldots, P_k\}$.

Clearly, we can rename the vertices so that we have $P_1 = \langle v_1, v_2, \ldots, v_n \rangle$. We will prove there exists a fairly large integer $1 \le m \le n$ and a set of integers $1 \le a_1 < a_2 < \ldots < a_m \le n$ such that, for every $1 \le i \le k$, $P_i[\{a_1, a_2, \ldots, a_m\}]$ is either $\langle v_{a_1}, v_{a_2}, \ldots, v_{a_m} \rangle$ or $\langle v_{a_m}, v_{a_{m-1}}, \ldots, v_{a_1} \rangle$. Then, we prove $G\left[\left\{v_{a_1}, v_{a_3}, \ldots, v_{a_{2\lceil \frac{m}{2} \rceil - 1}}\right\}\right]$ is an independent set.

We build a sequence $S = \langle s_1, s_2, \ldots, s_n \rangle$ of $(k-1)$-tuples in the following way: the ith element of S is $s_i = (e_{i,2}, e_{i,3}, \ldots, e_{i,k})$, where $e_{i,j}$ is the number of elements before v_i in P_j. According to De Bruijn's Theorem, and because S has n tuples, S has a subsequence of $m = \left\lfloor (n-1)^{\frac{1}{2^{k-1}}} \right\rfloor + 1$ tuples $\langle s_{a_1}, s_{a_2}, \ldots, s_{a_m} \rangle$ such that $\langle e_{a_1,j}, e_{a_2,j}, \ldots, e_{a_m,j} \rangle$ is monotonic for every $2 \le j \le k$. As a result, for every $1 \le x < y \le \lceil \frac{m}{2} \rceil$ and every $2 \le j \le k$, $v_{a_{2x}}$ is between $v_{a_{2x-1}}$ and $v_{a_{2y-1}}$ in P_j. Also, since $a_{2x-1} < a_{2x} < a_{2y-1}$, we know that $v_{a_{2x}}$ is between $v_{a_{2x-1}}$ and $v_{a_{2y-1}}$ in P_1. Hence, $v_{a_{2x}}$ is an obstacle to $\left\{v_{a_{2x-1}}, v_{a_{2y-1}}\right\}$. Hence, for every $1 \le x < y \le \lceil \frac{m}{2} \rceil$, $\left\{v_{a_{2x-1}}, v_{a_{2y-1}}\right\}$ is not $\mathcal{P}$-constructible, due to Definition 2.3.14. Consequently, for every $1 \le x < y \le \lceil \frac{m}{2} \rceil$, A does not have an edge between $v_{a_{2x-1}}$ and $v_{a_{2y-1}}$, due to Definition 2.3.16. So, $\left\{v_{a_1}, v_{a_3}, \ldots, v_{a_{2\lceil \frac{m}{2} \rceil - 1}}\right\}$ induces an independent set of size $\lceil \frac{m}{2} \rceil = \left\lfloor \frac{1}{2}(n-1)^{\frac{1}{2^{k-1}}} \right\rfloor + 1$ in A. □

We can use the fact that k-PC graphs have large independent sets to develop a greedy algorithm to color a k-PC graph.

Corollary 5.3.3. *For all n-vertex k-PC graphs A, $\chi(A) \in O(n^{1-\frac{1}{2^{k-1}}})$.*

Proof. According to Lemma 5.3.2, A has an independent set of size at least $\left\lfloor \frac{1}{2}(n-1)^{\frac{1}{2^{k-1}}} \right\rfloor + 1$. We color the vertices of this independent set with color one and remove them from A, obtaining a smaller graph A_1 with $n_1 < n - \left\lfloor \frac{1}{2}(n-1)^{\frac{1}{2^{k-1}}} \right\rfloor - 1$ vertices. Due to Lemma 5.2.1, A_1 is a subgraph of an n_1-vertex k-PC graph. Therefore, A_1 also has an independent set of size at least $\left\lfloor \frac{1}{2}(n_1-1)^{\frac{1}{2^{k-1}}} \right\rfloor + 1$. We continue to color every independent set found in this way, until the vertices of A has been partitioned into c independent sets $I_1, I_2, \ldots, I_c$.

To bound the number of independent sets obtained above, we consider the minimum number t such that $\sum_{i=1}^{t} |I_i| > \frac{n}{2} - 1$. For each $1 \le i \le t$, we have $|I_i| \ge \left\lfloor \frac{1}{2}(\frac{n}{2})^{\frac{1}{2^{k-1}}} \right\rfloor + 1$. Therefore, $t(\left\lfloor \frac{1}{2}(\frac{n}{2})^{\frac{1}{2^{k-1}}} \right\rfloor + 1) \le \sum_{i=1}^{t} |I_i| \le n$ and thus $t \le n/(\left\lfloor \frac{1}{2}(\frac{n}{2})^{\frac{1}{2^{k-1}}} \right\rfloor + 1) \le 2^{1+\frac{1}{2^{k-1}}} n^{1-\frac{1}{2^{k-1}}}$. Let $f(k,n)$ be the maximum number of independent sets obtained by the above greedy algorithm on an n-vertex k-PC graph. Clearly, $\chi(A) \leq c \leq f(k,n)$. Also, since the above argument shows that after removing vertices of the first $2^{1+\frac{1}{2^{k-1}}} n^{1-\frac{1}{2^{k-1}}}$ independent sets at most $\lfloor \frac{n}{2} \rfloor + 1$ vertices remain in the graph, we have the following recursive inequality for $n \geq 4$

$$f(k,n) \leq 2^{1+\frac{1}{2^{k-1}}} n^{1-\frac{1}{2^{k-1}}} + f\left(k, \left\lfloor \frac{n}{2} \right\rfloor + 1\right) \leq 2^{1+\frac{1}{2^{k-1}}} n^{1-\frac{1}{2^{k-1}}} + f\left(k, \frac{3}{4}n\right).$$

By solving the above inequality with the base case $f(k,1) = 1$, $f(k,2) = 2$, and $f(k,3) = 3$ we can conclude

$$\begin{aligned} f(k,n) &\leq 2^{1+\frac{1}{2^{k-1}}} n^{1-\frac{1}{2^{k-1}}} \left(1 + \left(\frac{3}{4}\right)^{1-\frac{1}{2^{k-1}}} + \left(\frac{3}{4}\right)^{2\left(1-\frac{1}{2^{k-1}}\right)} + \dots\right) \\ &= \frac{2^{1+\frac{1}{2^{k-1}}}}{1-\left(\frac{3}{4}\right)^{1-\frac{1}{2^{k-1}}}} n^{1-\frac{1}{2^{k-1}}}. \end{aligned}$$

□

5.4 Concluding Remarks

Since the size of the maximum clique of a graph is a lower bound on its chromatic number, it seems natural to ask how large the cliques of a k-PC graph can be. Using Lemma 5.3.2 it is simple to find an upper bound on the maximum clique size of k-PC graphs.

LEMMA 5.4.1. *A k-PC graph does not have a clique of size* $2^{2^{k-1}} + 1$.

Proof. Due to Lemma 5.2.1, if a k-PC graph has a clique of size $2^{2^{k-1}} + 1$, then $K_{2^{2^{k-1}}+1}$ is also a k-PC graph. Hence, it is enough to prove that $K_{2^{2^{k-1}}+1}$ cannot be a k-PC graph.

Suppose A is a complete graph on $2^{2^{k-1}} + 1$ vertices and A is a k-PC graph. According to Lemma 5.3.2, A has an independent set of size at least $\left\lfloor \frac{1}{2}(2^{2^{k-1}})^{\frac{1}{2^{k-1}}} \right\rfloor + 1 = 2$, which is a contradiction. □

In the proof of Lemma 5.3.2, instead of proving there exist three vertices x, y, and z such that y is an obstacle to $\{x,z\}$ we proved something stronger: we proved there exist three vertices x, y, and z such that $P_i[\{x,y,z\}]$ is always of the form $\langle x,y,z\rangle$ or $\langle z,y,x\rangle$. However, being an obstacle to $\{x,z\}$ means y never appears before both x and z, i.e. $P_i[\{x,y,z\}]$ is always of the form $\langle x,y,z\rangle$, $\langle x,z,y\rangle$, $\langle z,y,x\rangle$, or $\langle z,x,y\rangle$, where $P_i[\{x,y,z\}]$ is the permutation on $\{x,y,z\}$ imposed by P_i. By considering this fact, we may obtain a better upper bound on the clique size of k-PC graphs.

We can rephrase De Bruijn's Theorem in different words. A sequence S *avoids* a permutation $\langle p_1, p_2, \ldots, p_t\rangle$ of $\{1, 2, \ldots, t\}$ if S has no subsequence $\langle s_1, s_2, \ldots, s_t\rangle$ such that $s_i < s_j$ if and only if $p_i < p_j$. The permutation $\langle p_1, p_2, \ldots, p_t\rangle$ is called *a pattern*. One can generalize the above definition for sequences of k-tuples: a sequence S of k-tuples *avoids* a permutation P if the sequence of the ith elements of k-tuples in S avoids P, for every $1 \le i \le k$. So, De Bruin's Theorem states that every sequence of $n^{2^k} + 1$ k-tuples of integers has a subsequence S of length $n+1$ such that S avoids $\langle 1,3,2\rangle$, $\langle 2,1,3\rangle$, $\langle 2,3,1\rangle$, and $\langle 3,1,2\rangle$ for every $1 \le i \le k$. We refer the reader to the papers on sequences that avoid a given pattern [14, 7] or a given set of patterns [80] for more details.

The case that we are interested in has not been previously studied. Suppose $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k)$ is the smallest number such that every sequence of $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k)$ k-tuples of integers has a subsequence of length $n+1$ avoiding $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$. We want to investigate how much smaller $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k)$ is than $n^{2^k} + 1$. $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k)$ may be smaller than $n^{2^k} + 1$, because any monotonic sequence avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$, but the reverse is not true. For example, $\langle 8,7,1,6,5,2,4,3\rangle$ is not ascending nor is descending, but avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$.

In the following lemma, we prove if a sequence avoids $\langle 2,3,1\rangle$ and $\langle 2,1,3\rangle$, then it has a large monotonic subsequence. This helps us to prove that $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k)$ is not much smaller than n^{2^k}, and thus, we can conclude that refining De Bruijn's Theorem for our purpose cannot lead to a significantly better bound.

Lemma 5.4.2. *If P is a permutation of length at least $2n+1$ and P avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$, then P has a monotonic subsequence of length at least $n+1$.*

Proof. Suppose $P = \langle p_1, p_2, \ldots, p_{2n+1}\rangle$ is a permutation, and P avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$. Then, $M = \{p_i : \forall j > i \quad p_j > p_i\}$ is a subset of $\{1, 2, \ldots, 2n+1\}$ and $P[M]$ is an increasing subsequence of P. If M has more than n elements, then the lemma holds for P. Otherwise, assume that M has $a \le n$ elements. We denote the ith element of $P[M]$ by m_i. For example, in Figure 5.4, $a = 3$ and $M = \{p_4, p_7, p_9\}$. Hence, $m_1 = p_4$, $m_2 = p_7$, and $m_3 = p_9$.

Elements of M divide P into a segments $S_1, S_2, \ldots, S_a$; more precisely, we define

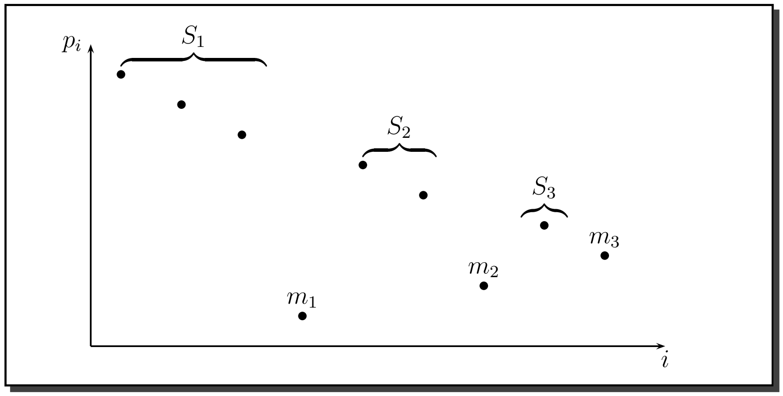

Figure 5.4: A permutation that avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$.

$S_i = \{p_j : p_j$ is before m_i and after m_{i-1} in $P\}$ for all $2 \leq i \leq a$, and we denote the set of all elements of P that are before m_1 in P by S_1. Note that $m_a = p_{2n+1}$ and thus there is no element after m_a in P. The way we chose elements of M ensures that no number in S_i is less than m_i. Otherwise, suppose an element of S_i is less than m_i. Also, suppose that s is the last element of S_i that is less than m_i. The fact that all the elements between s and m_i are at least m_i, together with the fact that m_i is less than all the elements of P after m_i implies that s is less than all the elements of P after s. This means that s is in M, which is a contradiction.

Now we build a decreasing sequence of size $n + 1$. Each S_i induces a decreasing subsequence in P because if p_x appears before p_y in $P[S_i]$ and $p_x < p_y$, then $P[\{p_x, p_y, m_i\}]$ is a $\langle 2,3,1\rangle$-pattern. For example, in Figure 5.4, if $p_1 < p_3$, $P[\{p_1, p_3, m_1\}]$ would be a $\langle 2,3,1\rangle$-pattern. On the other hand, for any $1 \leq i < j \leq a$ the last element of $P[S_i]$ is greater than the first element of $P[S_j]$, as otherwise these two elements together with m_i induce a $\langle 2,1,3\rangle$-pattern in P. Therefore, $P[\cup_{i=1}^{a} S_i]$ is a decreasing subsequence of P. Note that, $M \cup (\cup_{i=1}^{a} S_i) = \{1, 2, \ldots, 2n+1\}$ and $|M| \leq n$. Hence, $| \cup_{i=1}^{a} S_i| \geq n + 1$. □

Using Lemma 5.4.2 it is easy to prove that $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n, k)$ is close to n^{2^k}.

Corollary 5.4.3. *For all integers n and k we have* $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n, k) \geq \left(\left\lfloor \frac{n}{2^k} \right\rfloor\right)^{2^k} + 1$.

Proof. We use the tightness of De Bruijn's bound: we know there exists a sequence of n^{2^k} k-tuples that have no subsequence S of length $n + 1$ such that the sequence of the ith elements of k-tuples in S is monotonic for every $1 \leq i \leq k$.

Suppose, for the sake of contradiction, that $f_{\langle 2,1,3\rangle,\langle 2,3,1\rangle}(n,k) \leq \left(\lfloor \frac{n}{2^k} \rfloor\right)^{2^k}$. Then, every sequence of n^{2^k} k-tuples has a subsequence S_1 of length $2^k n+1$ such that the sequence of the ith elements of k-tuples in S_1 avoids $\langle 2,1,3\rangle$ and $\langle 2,3,1\rangle$ (for every $1 \leq i \leq k$). According to Lemma 5.4.2, S_1 has a subsequence S_2 of length $2^{k-1}n+1$ such that, say, the sequence of the first elements of k-tuples in S_2 is monotonic. Similarly, S_2 has a subsequence S_3 of length $2^{k-2}n+1$ such that the sequences of, say, the first and second elements of k-tuples in S_3 are both monotonic, and so on. Finally we end up with a subsequence S_{k+1} of length $n+1$ such that the sequence of the ith elements of k-tuples in S_{k+1} is monotonic for all $1 \leq i \leq k$, which is a contradiction. □

In fact, using a probabilistic argument we can prove there exists a k-PC graph that has a clique of size $\Omega(1.2247^k)$.

Lemma 5.4.4. *$K_{\lfloor 0.4952 \cdot 1.2247^k \rfloor - 1}$ is a k-PC graph.*

Proof. Suppose $\mathcal{P}$ is a set of k random permutations P_1, P_2, ..., P_k on $V = \{v_1, v_2, \ldots, v_n\}$. We use $E_{x,y,z}$ to denote the event that v_z is an obstacle to $\{v_x, v_y\}$ for every three mutually nonidentical vertices v_x, v_y, and v_z. Since each permutation is chosen randomly, independent of the other permutations, the probability of $E_{x,y,z}$ occurring is $(\frac{2}{3})^k$, because with probability $\frac{2}{3}$ z is before one of x or y in each permutation. Because there are $\binom{n}{3}$ events $E_{x,y,z}$, the probability of at least one of these events occurring is at most $\binom{n}{3}(\frac{2}{3})^k$. Therefore, if $\binom{n}{3}(\frac{2}{3})^k < 1$, we can conclude there exists a set of k permutations $\mathcal{P}^*$ on V such that the graph constructed on $\mathcal{P}^*$ is a clique.

The above argument gives us a lower bound of $\Omega((\frac{3}{2})^{\frac{k}{3}} \approx 1.1447^k)$. However, we can improve it using the Lovász Local Lemma [30]. The following is known as the symmetric case of Theorem 3.1.8.

Lemma 5.4.5. **The Local Lemma, Symmetric Case** *[30]. Suppose A_1, A_2, ..., and A_n are random events such that for every $1 \leq i \leq n$ A_i occurs with probability at most p and A_i is mutually independent of all but at most d other events. Then, if $ep(d+1) \leq 1$, the probability that none of A_i's occurs is more than zero.*

The key point is that E_{x_1,y_1,z_1} is not dependent on E_{x_2,y_2,z_2} if $\{x_1, y_1, z_1\}$ and $\{x_2, y_2, z_2\}$ are disjoint. Therefore, each event E_{x_1,y_1,z_1} is dependent on at most $3n + 3\binom{n}{2}$ other events: $3n$ events E_{x_2,y_2,z_2} such that $\{x_1, y_1, z_1\} \cap \{x_2, y_2, z_2\} = 2$ and $3\binom{n}{2}$ events E_{x_2,y_2,z_2} such that $\{x_1, y_1, z_1\} \cap \{x_2, y_2, z_2\} = 2$. Hence, if $e(\frac{2}{3})^k(3n + 3\binom{n}{2} + 1)$ is less than one, the probability that none of the events occurs is more than zero. Thus, if $n = \lfloor 0.4952 \cdot 1.2247^k \rfloor - 1$, $e(\frac{2}{3})^k(3n+3\binom{n}{2}+1)$ will be less than one, and therefore, there exists a set of k permutations $\mathcal{P}^*$ on $V = \{v_1, v_2, \ldots, v_n\}$ such that the graph constructed on $\mathcal{P}^*$ is a clique. $\square$

Note that, Lemma 5.4.4 proves there exist k-PC graphs whose chromatic numbers are at least $\lfloor 0.4952 \cdot 1.2247^k \rfloor - 1$. By combining this fact with Theorem 5.1.5 we can prove that the chromatic number of a hypergraph induced by n k-dimensional axis-parallel boxes can be as large as $\Omega(1.2247^k)$, which is exponential with respect to k.

Chapter 6

Conclusion

We generalized the hypergraph coloring problem and introduced the intersperse coloring problem. We showed that many coloring problems such as strong coloring of hypergraphs, the star coloring problem, the problem of proper coloring of graph powers, the acyclic coloring problem, and the frugal coloring problem are special cases of the intersperse coloring problem.

In Theorem 3.1.2, using a probabilistic method, we obtained an upper bound of

$$c + c\sum_{E\in\mathcal{E}(N)}\sum_{i=1}^{\min\{|E|,\ell\}-1} p^{(\frac{1}{c})}_{|E|,\frac{i}{\min\{|E|,\ell\}-1}}$$

for hypergraphs in the general case, where N is a hypergraph, $c > 0$ is any constant integer, and $p^{(\alpha)}_{n,\delta}$ is defined in Notation 3.1.1. Then, we used the method of conditional expectations to develop the deterministic version of Theorem 3.1.2 in Theorem 3.1.5. This result works for all coloring problems that can be viewed as special cases of the intersperse coloring problem. For example, Theorem 3.1.5 gives a polynomial-time deterministic algorithm for properly coloring r-uniform hypergraphs with m edges using at most $O(m^{\frac{1}{r}})$ colors. If we apply Theorem 3.1.5 to the strong coloring problem of hypergraphs with few large hyperedges, we get a better upper bound than the one in Agnarsson and Halldórsson's [3] work. Agnarsson and Halldórsson's bound for a hypergraph N of rank r is $r\sqrt{|\mathcal{E}(N)|}$, which is independent of the number of large hyperedges. This is a good bound for the general case; however, in an example in Chapter 3 we showed that if the number of large hyperedges is much less than the number of small hyperedges, the above-mentioned bound is not efficient. We proved that if only x hyperedges of N are of size r and the other hyperedges are of size two, then the strong chromatic number is at most $\sqrt{(|\mathcal{E}(N)| - x)(xr - x + 1)}$.

We also obtained two upper bounds, one constructive and one existential, as functions of vertex degrees of the input hypergraph. In Theorem 3.1.6, we proved that there is a polynomial-time algorithm that finds an ℓ-intersperse coloring of a hypergraph N with at most $\max_{v \in V(N)} \left\{ \sum_{E \in \mathcal{E}(N) \text{ s.t. } v \in E} (\min \{|E|, \ell\} - 1) \right\} + 1$ colors. The existential bound, which was proven in Theorem 3.1.10 with the help of the local lemma, is $\left\lfloor \max_{\ell \leq i \leq |V(N)|} \left\{ 2(\ell - 1) \left(4e^{\ell-1} f(i)\right)^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1$, where $f(i)$ is the maximum number of hyperedges of size i intersecting with a single hyperedge.

We considered two special families of hypergraphs: copy hypergraphs and geometric hypergraphs induced by d-dimensional boxes. Copy hypergraphs are introduced in this thesis for the first time and we showed that the intersperse coloring problem on copy hypergraphs is an interesting problem, because it covers many other coloring problems. We obtained two upper bounds for $\mathrm{SC}(A, \mathcal{B}, \ell)$, where A is a graph, $\mathcal{B}$ is a finite family of graphs, and $\ell \geq 2$ is an integer. The first upper bound is $(\ell - 1) \cdot \sum_{B \in \mathcal{B}} \left(|V(B)| \cdot \Delta(A)^{|V(B)|-1}\right) + 1$ and the second upper bound is $\left\lfloor 2(\ell - 1) \max_{\ell \leq i \leq |V(A)|} \left\{ (cib_i \Delta(A)^{i-1})^{\frac{1}{i-\ell+1}} \right\} \right\rfloor + 1$. In the second bound it is assumed that $\ell \leq \min_{B \in \mathcal{B}} |V(B)|$. For the case that $\ell = 2$ we obtained a third upper bound that is stronger than the previous two upper bounds for some instances. Also, in Theorem 4.2.11, we obtained a lower bound for the case that ℓ can be any number, but $\mathcal{B}$ is a family of trees. The lower bound is $\max_{B \in \mathcal{B}} \frac{2|\mathcal{E}(A)|(\min\{|V(B)|, \ell\} - 2)}{(1+\varepsilon)(|V(B)|-1)|V(A)|} + 1$, which is true for any value of $\varepsilon > 0$ if all trees in $\mathcal{B}$ are large enough.

In this thesis, we also proposed a new model for graph and hypergraph property testing, called the symmetric model. The symmetric model is the first model that can be used for developing property testing algorithms for non-uniform hypergraphs. We proved that there are a number of graph properties, in particular P_2-freeness, that have efficient property testers in the symmetric model but do not have any efficient property tester in previously-known property testing models. Note that the problem of checking whether a given graph G is P_2-free is equivalent to the problem of checking whether $\mathrm{ISC}(G, \{P_2\}, 2) = 1$. So, it is a special case of the intersperse coloring problem on induced copy hypergraphs.

Future Work

In Chapter 3 we proposed a simple $O(\frac{n}{\lg n})$-approximation algorithm for properly coloring c-colorable hypergraphs, for any integer c. However, it is not clear if the simple method used can be generalized to work for the intersperse coloring problem. Thus, finding an $O(\frac{n}{\lg n})$-approximation algorithm for the intersperse coloring problem on c-colorable hypergraphs remains open.

Open Problem 6.1. *Is there an* $O(\frac{n}{\lg n})$*-approximation algorithm for the* ℓ*-intersperse coloring problem on* c*-colorable non-uniform hypergraphs, for any constant* c*?*

The above discussion was regarding general hypergraphs. As a special family of hypergraphs, one can consider the class of neighborhood hypergraphs. Neighborhood hypergraphs, introduced by Chastel et al. [24], and neighborhood hypergraphs of radius r, first introduced in this thesis, are interesting to study from the intersperse coloring problem point of view, because, as mentioned in Chapter 2, several facility location problems can be modelled in this way.

Another area that may be a fruitful direction for future work is studying the property testing algorithm under the symmetric model, defined in this thesis. As mentioned above, we proposed a property tester for P_2-freeness in the symmetric model that uses a constant number of queries, independent of the size of the input graph. However, it is not clear for what connected graphs B we can test B-freeness. In particular, the following problem remains open:

Open Problem 6.2. *Does there exist a property testing algorithm for* B*-freeness using only* $O(\mathrm{poly}(\frac{1}{\varepsilon}))$ *queries, for every connected graph* B*?*

For geometric hypergraphs, we considered the proper coloring problem only on geometric hypergraphs induced by axis-parallel boxes. Naturally, one can think about the intersperse coloring problem on geometric hypergraphs. However, the problem still has room for more work, even in the proper coloring context. For example, we reduced the proper coloring of geometric hypergraphs induced by axes-parallel boxes to the coloring of k-PC graphs. But, the best upper bound for the chromatic number of k-PC graphs we could find was of the form $O(n^{1-f(k)})$, where f is a decreasing function of k, for $k \geq 4$. Hence, another open problem is the following:

Open Problem 6.3. *Is it true that, for any constant integer number* $k \geq 4$ *and real number* $\varepsilon > 0$*, the chromatic number of* k*-PC graphs is in* $o(n^{\varepsilon})$*?*

Since the intersperse coloring problem is a broad problem, there are lots of related topics that are not covered in this thesis and are left as future areas of study. This thesis should be mostly viewed as an introduction to the intersperse coloring problem and why it is interesting.

Bibliography

[1] G. Agnarsson, R. Greenlaw, and M. Halldórsson. On powers of chordal graphs and their colorings. *Congressus Numerantium*, 100:41–65, 2000.

[2] G. Agnarsson and M. Halldórsson. Coloring powers of planar graphs. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 654–662, 2000.

[3] G. Agnarsson and M. M. Halldórsson. Strong colorings of hypergraphs. In *Proceedings of the Second Workshop on Approximation and Online Algorithms*, pages 253–266, 2004.

[4] M. Ajtai, J. Komlós, and E. Szemerédi. On the Erdös-Sós conjecture - the dense case. *Manuscript*, 1991.

[5] N. Alon. Testing subgraphs in large graphs. *Random Structures and Algorithms*, 21(3–4):359–370, 2002.

[6] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 645–655, 1999.

[7] N. Alon and E. Friedgut. On the number of permutations avoiding a given pattern. *Journal of Combinatorial Theory Series A*, 89(1):133–140, 2000.

[8] N. Alon, P. Kelsen, S. Mahajan, and H. Ramesh. Approximate hypergraph coloring. *Nordic Journal of Computing*, 3:425–439, 1996.

[9] N. Alon, M. Krivelevich, and B. Sudakov. Turán numbers of bipartite graphs and related Ramsey-type questions. *Combinatorics, Probability and Computing*, 12, 2005.

[10] N. Alon and B. Mohar. The chromatic number of graph powers. *Combinatorics, Probability and Computing*, 11:1–10, 2002.

[11] N. Alon and J.H. Spencer. *The Probabilistic Method*. John Wiley and Sons, Inc., New York, 1992.

[12] N. Alon, R. Yuster, and U. Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. In *Proceedings of the 26th Symposium on Theory of Computing*, pages 326–335, 1994.

[13] M. Arai, M. Yamashita, T. Hirata, T. Ibaraki, and N. Honda. Lower bounds on the time complexity for some subgraph detection problems. *Systems and Computers in Japan*, 17:95–102, 1986.

[14] R. Arratia. On the Stanley-Wilf conjecture for the number of permutations avoiding a given pattern. *Electronic Journal of Combinatorics*, 6:N1, 1999.

[15] J. Beck. An algorithmic approach to the Lovász Local Lemma. *Random Structures and Algorithms*, 2:343–365, 1991.

[16] C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. Ullman, and M. Yannakakis. Properties of acyclic database schemes. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, pages 355–362, 1981.

[17] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30(3):479–513, 1983.

[18] C. Berge. *Hypergraphs*. North Holland, Amsterdam/New York, 1989.

[19] B Bollobás. Extremal graph theory. In *Handbook of Combinatorics (volume 2)*, pages 1231–1292, 1995.

[20] O. V. Borodin. On acyclic colorings of planar graphs. *Discrete Mathematics*, 25:211–236, 1979.

[21] J. I. Brown and D. G. Corneil. On generalized graph colorings. *Journal of Graph Theory*, 11:87–99, 1987.

[22] W. G. Brown and M. Simonovits. Digraph extremal problems, hypergraph extremal problems, and the densities of graph structures. *Discrete Mathematics*, 48(2-3), 1984.

[23] M. I. Burstein. Every 4-valent graph has an acyclic 5 coloring (in russian). *Soobšč. Akademiia Nauk Gruzin*, 93:21–24, 1979.

[24] S. Chastel, P. Colantoni, and A. Bretto. Displaying image neighborhood hypergraphs line-graphs. In *Proceedings of the Tenth International Conference on Discrete Geometry for Computer Imagery*, pages 124–135, 2002.

[25] H. Chen and A. M. Frieze. Coloring bipartite hypergraphs. In *Proceedings of the Fifth International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 345–358, 1996.

[26] V. Chvátal and E. Szemerédi. On the Erdös-Stone theorem. *Journal of the London Mathematical Society*, 23:207–214, 1981.

[27] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGraw-Hill, New York, 1990.

[28] A. Czumaj and C. Scheideler. Coloring non-uniform hypergraphs: a new algorithmic approach to the general lovász local lemma. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete algorithms*, pages 30–39, 2000.

[29] P. Erdös and T. Gallai. On maximal paths and circuits of graphs. *Acta Mathematica Hungarica*, 10(3–4):337–356, 1959.

[30] P. Erdös and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *(A. Hajnal et al. eds) Finite and Infinite Sets: Colloquia Mathematica Societatis Janos Bolyai*, pages 609–627, 1975.

[31] P. Erdös and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3):298–301, 1973.

[32] P. Erdös and M. Simonovits. A limit theorem in graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 1, 1966.

[33] P. Erdös and V. T. Sós. Extremal problems in graph theory. In *Proceedings of the Symposium held in Smolenice*, pages 29–38, 1963.

[34] P. Erdös and G. Szekeres. A combinatorial theorem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[35] A. M. Farley. Minimal path broadcast networks. *Networks*, 43(2):61–70, 2004.

[36] G. Fertin, E. Godard, and A. Raspaud. Acyclic and k-distance coloring of the grid. *Information Processing Letters*, 87(1):51–58, 2003.

[37] G. Fertin and A. Raspaud. Acyclic coloring of graphs with maximum degree Δ. In *Proceedings of the Third Euroconference on Combinatorics, Graph Theory and Applications*, pages 389–396, 2005.

[38] G. Fertin, A. Raspaud, and B. Reed. On star coloring of graphs. In *Proceedings of the Twenty Seventh International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 140–153, 2001.

[39] F. V. Fomin, F. Grandoni, A. Pyatkin, and A. Stepanov. Bounding the number of minimal dominating sets: a measure and conquer approach. In *Proceedings of the Sixtieth Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827, pages 573–582, 2005.

[40] F. V. Fomin, D. Kratsch, and G. J. Woeginger. Exact (exponential) algorithms for the dominating set problem. In *Proceedings of the Thirtieth International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 245–256, 2004.

[41] Z. Füredi. Turán type problems. In *(A. D. Keedwell ed.) Proceedings of the Thirteenth British Combinatorial Conference*, 1991.

[42] M. R. Garey and D. S. Johnson. *Computers and Intractability*. Freeman, 1979.

[43] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[44] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing*, pages 406–415, 1997.

[45] B. Grünbaum. Acyclic colorings of planar graphs. *Israel Journal of Mathematics*, 14(3), 1973.

[46] V. Guruswami, J. Hastad, and M. Sudan. Hardness of approximate hypergraph coloring. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 149–158, 2000.

[47] F. Harary. Conditional colorability in graphs. In *Proceedings of the First Colorado Symposium on Graph Theory*, pages 127–136, 1985.

[48] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[49] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs: Advanced Topics*. Marcel Dekker, New York, 1998.

[50] H. Hind, M. Molloy, and B. Reed. Colouring a graph frugally. *Combinatorica*, 17(4):469–482, 1997.

[51] H. Hind, M. Molly, and B. Reed. Total colouring with $\delta+$ polylog(δ) colours. *SIAM Journal on Computing*, 28:816–821, 1998.

[52] T. Hofmeister and H. Lefmann. Approximating maximum independent sets in uniform hypergraphs. In *Proceedings of the Twenty-Third International Symposium on Mathematical Foundations of Computer Science*, pages 562–570, 1998.

[53] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1442–1483, 2004.

[54] S. Khot. Hardness results for approximate hypergraph coloring. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 351–359, 2002.

[55] S. Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 23–32, 2002.

[56] J. Komlós and M. Simonovits. Szemerédi's regularity lemma and its applications in graph theory. In *(D. Miklós, V. T. Sós, T. Szónyi eds.) Bolyai Society Mathematical Studies 2, Combinatorics, Paul Erdós is Eighty*, volume 2, pages 295–352, 1996.

[57] A. Kostochka, D. Mubayi, V. Rödl, and P. Tetali. On the chromatic number of set systems. *Random Structures and Algorithms*, 19(2):87–98, 2001.

[58] M. Krivelevich, R. Nathaniel, and B. Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *Journal of Algorithms*, 41(1):99–113, 2001.

[59] M. Krivelevich and B. Sudakov. Approximate coloring of uniform hypergraphs. *Journal of Algorithms*, 49(1):2–12, 2003.

[60] J. B. Kruskal. Monotonic subsequences. *Proceedings of the American Mathematical Society*, 4(2):264–274, 1953.

[61] G. M. Kuper. An algorithm for reducing acyclic hypergraphs. Technical report, Stanford University, Stanford, CA, USA, 1982.

[62] L. Lovász. On decomposition of graphs. *Studia Scientiarum Mathematicarum Hungarica*, 1:237–238, 1966.

[63] L. Lovász. Colorings and coverings of hypergraphs. In *Proceedings of the Fourth South-Eastern Conference on Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.

[64] D. Maier. *Theory of Relational Databases*. Computer Science Press, Inc., Rockville (Maryland), 1983.

[65] D. Maier and J. D. Ullman. Connections in acyclic hypergraphs: Extended abstract. In *Proceedings of the First ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 34–39, 1982.

[66] S. T. McCormick. Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem. *Mathematical Programming*, 26(2):153–171, 1983.

[67] M. Molloy and B. Reed. Further algorithmic aspects of the local lemma. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 524–529, 1998.

[68] M. Molly and M. R. Salavatipour. Frequency channel assignment on planar networkscoloring powers of planar graphs. In *Proceedings of the Tenth Annual European Symposium on Algorithms*, pages 736–747, 2002.

[69] A. M. Odlyzko, J. B. Shearer, and R. C. Siders. Monotonic subsequences in dimensions higher than one. *Electronic Journal of Combinatorics*, 4(2):R14, 1997.

[70] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2):165–183, 2002.

[71] S. Pemmaraju and A. Srinivasan. Improved bounds and algorithms for frugal coloring. *http://www.cs.uiowa.edu/ rraman/frugal.pdf*, accessed in 2007.

[72] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[73] M. R. Salavatipour. A $(1 + \epsilon)$-approximation algorithm for partitioning hypergraphs using a new algorithmic version of the lovász local lemma. *Random Structructures and Algorithms*, 25(1):68–90, 2004.

[74] N. Simjour. *A New Optimality Measure for Distance Dominating Sets*. Master's Thesis, School of Computer Science, University of Waterloo, Waterloo, 2006.

[75] S. S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, New York, 1997.

[76] S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *Proceedings of the ACM/SIAM Symposium on Discrete Algorithms*, pages 316–323, 2006.

[77] E. Szemerédi. Regular partitions of graphs. In *Colloques Internationaux C.N.R.S. N-260-Problémes Combinatoires et Théorie des Graphes*, pages 399–401, 1976.

[78] M. Trick. Network resources for coloring a graph. *http://mat.gsia.cmu.edu/COLOR/color.html*, accessed in 2007.

[79] P. Turán. On an extremal problem in graph theory (in Hungarian). *Matematikai és Fizikai Lapok*, 48:436–452, 1941.

[80] V. R. Vatter. Permutations avoiding two patterns of length three. *Electronic Journal of Combinatorics*, 9:R6, 2003.

[81] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.

[82] G. Wegner. Graphs with a given diameter and a coloring problem. *Technical Report, University of Dortmund*, 1977.

[83] D. B. West. *Introduction to Graph Theory, Second Edition*. Prentice-Hall, Englewood Cliffs, NJ, 2000.

[84] D. R. Wood. Acyclic, star and oriented colourings of graph subdivisions. *Discrete Mathematics and Theoretical Computer Science*, 7(1):37–50, 2005.